SYHUNT

Documentação do Syhunt

Guia do Usuário e de Integração

Versão 7.2

Documentação do **Syhunt Hybrid** e de seus módulos: Syhunt Dynamic (DAST), Syhunt Code e Mobile (SAST/MAST), Syhunt API e Syhunt Forensic.

É novo no Syhunt? Comece pelo Guia de Boas-Vindas e, em seguida, pelo Início Rápido.

Gerado em 2026-10-26 · www.syhunt.com

Syhunt Hybrid 7.2

Sumário

PRIMEIROS PASSOS

Boas-Vindas ao Syhunt
Início Rápido
Download
Plano de Testes de Prova de Conceito
Perguntas Frequentes

INSTALAÇÃO

Visão Geral da Instalação e Windows
Instalação no Linux
Instalação no macOS
Interface Web
Instalando o Wine
Notas de Migração
Ajuste de Desempenho

VARREDURAS

Varredura Dinâmica (DAST)
Varredura Autenticada
Varredura de Código-Fonte (SAST/MAST)
Varredura de APIs Web
Métodos de Varredura
Regras de Exclusão
Agendamento de Varreduras
Análise Forense de Logs
Relatórios

INTEGRAÇÕES

Linha de Comando (CLI)
PowerShell
API Web
API Lua
GitHub
GitLab
Bitbucket
Azure DevOps
Jenkins
Rastreadores de Issues
DefectDojo
Faraday

REFERÊNCIA

Especificações Técnicas
Preferências
Verificações de Vulnerabilidades

Página 2 de 182

Syhunt Hybrid 7.2

SUPORTE

Como Relatar Bugs e Obter Suporte

Página 3 de 182

Syhunt Hybrid 7.2

Primeiros passos

Página 4 de 182

Syhunt Hybrid 7.2

Boas-Vindas ao Syhunt

Obrigado por escolher o Syhunt. Nosso objetivo é ajudar você a aproveitar ao máximo as ferramentas Syhunt, para que sua organização alcance mais rapidamente um nível mais alto de segurança de aplicações e de conformidade, encontrando e corrigindo vulnerabilidades e fraquezas antes que cheguem à produção.

Comece pelos quatro passos abaixo e depois leia o restante deste guia.

Passos iniciais

1. Ative sua licença

Se ainda não ativou sua licença, faça isso agora para começar a usar as ferramentas Syhunt.

1. Baixe o instalador pelo link fornecido pela Syhunt ou pelo seu Parceiro Autorizado Syhunt.
2. Clique duas vezes no arquivo baixado e siga o instalador (um processo padrão *Next* → *Next* → *Finish*). O Syhunt é aberto quando você clica em **Finish**.
3. Informe seu arquivo de chave do produto (Product Key File), seguindo as instruções do e-mail ou habilitando a interface web, que o solicitará. Use o arquivo de chave anexado ao e-mail que continha o link de download.
4. Clique em **Apply**. Uma mensagem de sucesso confirma que o Syhunt está pronto, e as opções disponíveis passam a ficar habilitadas na tela inicial.

Se o Syhunt não pedir uma chave, já existe uma licença ativa, que precisa ser substituída. Para substituí-la, abra `https://127.0.0.1:8018/syhunt/?n=license` ou execute novamente o comando da CLI que você usou para ativar sua chave.

2. Inscreva-se para receber notificações

Assine o boletim para clientes para receber um e-mail sempre que houver atualizações e melhorias disponíveis. As atualizações podem ser baixadas e instaladas diretamente do site da Syhunt. O Syhunt 6.8.4 e versões posteriores também anunciam novas versões no Launcher e nos relatórios.

- Inscreva-se aqui

3. Siga o plano de testes de prova de conceito

Recomendamos fortemente que você execute o Plano de Testes de Prova de Conceito do Syhunt. Ele conduz você por varreduras em diversas condições que simulam o uso real. Ao final, você estará familiarizado com os principais recursos do produto e com as opções de integração.

Página 5 de 182

Syhunt Hybrid 7.2

- Plano de testes (Markdown)
- Plano de testes (PDF, em inglês)

4. Leia os guias de início rápido

Reserve um tempo para que você e sua equipe técnica leiam os guias de início rápido e os demais documentos deste repositório, e se familiarizem com as ferramentas e recursos que sua organização pretende usar.

- Índice da documentação
- Área de Suporte da Syhunt

Como obter ajuda

A Área de Suporte da Syhunt é a forma mais rápida de encontrar respostas. Se tiver uma dúvida ou encontrar um problema, abra um ticket de suporte. Ao relatar um falso positivo ou qualquer outro problema, siga as orientações para relatar bugs para uma solução mais rápida.

Solicitação de recursos

Você pode ajudar a definir o roadmap do produto. Envie a qualquer momento pedidos de novos recursos e melhorias à nossa equipe de gestão de produto, abrindo um ticket de suporte.

Consulta e renovação da licença

A Syhunt ou seu Parceiro Autorizado Syhunt entrará em contato cerca de 90 dias antes do término da sua licença, para que você possa renovar a licença ou a cobertura de suporte. O Syhunt 6.7.1 e versões posteriores também exibem um lembrete na interface principal cerca de 60 dias antes da data de término. Para dúvidas sobre esse processo ou sobre necessidades adicionais, entre em contato com a Syhunt ou com seu Parceiro Autorizado Syhunt.

Você pode consultar o prazo e os detalhes da sua licença a qualquer momento em **Help** → **License**.

Treinamento

Pode haver treinamento disponível na sua região e no seu idioma. Entre em contato com seu Parceiro Autorizado Syhunt para verificar disponibilidade, valores e datas.

Por enquanto é isso. Para qualquer outra necessidade, fique à vontade para falar conosco.

Página 6 de 182

Syhunt Hybrid 7.2

Início Rápido

> **Aplica-se a:** Syhunt Hybrid 7.2

O Syhunt Hybrid é uma suíte multilinguagem de avaliação de segurança de aplicações web. Ele procura as falhas mais comuns em aplicações web sob a perspectiva de um atacante: injeta dados nas aplicações e analisa suas respostas para determinar se o código é vulnerável a ataques específicos, como injeção de SQL, cross-site scripting (XSS) e muitos outros. Quando solicitado, o Syhunt também faz a varredura do código-fonte da aplicação em busca de problemas de segurança.

O que o Syhunt pode verificar?

O Syhunt foi projetado para cobrir uma ampla variedade de plataformas:

- **Qualquer plataforma de servidor web** (por meio da varredura dinâmica), seja Unix, Linux ou Windows.
- **Dispositivos que servem páginas web**, como roteadores e firewalls.
- **Aplicações web escritas em** ASP.NET, Java, Node.js, Lua, Perl, PHP, Python, Ruby e muitas outras linguagens (por meio da varredura de código-fonte).
- **Firewalls de aplicação web e sistemas de detecção de intrusão** (por meio de suas técnicas de evasão).

Uma abordagem proativa

Scanners de aplicações web costumam ser vistos como antivírus: reativos e dependentes de atualizações frequentes de assinaturas, diárias ou pelo menos semanais. Isso vale para os scanners tradicionais. Desde 2008, porém, a Syhunt adota uma abordagem proativa na construção de suas verificações e de seu scanner híbrido, concentrando-se em fraquezas comuns (CWEs), e não em vulnerabilidades divulgadas individualmente (CVEs).

Scanners tradicionais (reativos)	A abordagem proativa do Syhunt
Dependem fortemente de atualizações regulares de assinaturas voltadas a CVEs (**vulnerabilidades conhecidas e divulgadas** em aplicações de terceiros).	Não depende de atualizações frequentes. O desenvolvimento se concentra em uma lista de CWEs (**fraquezas comuns**). Verificações específicas para CVEs são lançadas com as versões menores e maiores, como **complemento** às verificações proativas.
Não detectam vulnerabilidades não divulgadas em aplicações de terceiros.	Detecta **vulnerabilidades divulgadas e não divulgadas** em aplicações de terceiros: detectar com precisão uma única CWE significa identificar centenas de CVEs de uma só vez.
Deixam passar a maioria, se não todas, as vulnerabilidades das suas aplicações próprias.	É muito mais adequado para encontrar vulnerabilidades nas suas aplicações próprias.
Não analisam código-fonte.	Realiza **análise híbrida**: a análise dinâmica complementa a análise de código-fonte (DAST + SAST).

Próximos passos

Tarefa	Guia
Executar sua primeira varredura dinâmica (caixa-preta)	Varredura Dinâmica
Executar sua primeira varredura de código-fonte (caixa-branca)	Varredura de Código-Fonte
Escolher o método de varredura adequado	Métodos de Varredura
Agendar varreduras recorrentes	Agendamento de Varreduras
Personalizar relatórios	Relatórios
Verificar o que é necessário antes de uma varredura dinâmica	Requisitos prévios

Página 7 de 182

Syhunt Hybrid 7.2

Tarefa	Guia
Verificar os requisitos de hardware e software	Especificações Técnicas

Redefinindo a severidade de uma vulnerabilidade

Se você discordar da severidade atribuída a uma vulnerabilidade relatada, pode definir uma nova severidade, que valerá para todas as varreduras futuras:

1. Na interface web, abra a tela Connectors (**Menu** → **Connectors**).
2. Adicione um conector do tipo **List** chamado `VULNSEVERITY`.
3. Informe o ID único da verificação da vulnerabilidade, seguido de `=` e da nova severidade. Por exemplo: `C-1603660271-7241=low`
4. Clique em **Add** para adicionar a entrada à lista.
5. Clique em **Save** para salvar a lista.

Página 8 de 182

Download

Os instaladores do Syhunt são distribuídos pela página de downloads do Syhunt.

Edição	Como obter
Syhunt Hybrid e **Syhunt Runner**	Use o link de download e a chave de cliente enviados pela Syhunt ou pelo seu Parceiro Autorizado Syhunt.
Demonstração / avaliação	Use o link de download de avaliação recebido após solicitar uma demonstração.
Syhunt Community	Disponível na página de downloads, sem chave.
Prey Server (aplicações de teste vulneráveis)	Consulte Prey Server no plano de testes de prova de conceito.

Se a sua licença expirou, entre em contato com a Syhunt para renová-la.

Com o instalador em mãos, siga o guia da sua plataforma:

- Windows
- Linux
- macOS
- Wine (Linux e macOS)

Syhunt Hybrid 7.2

Plano de Testes de Prova de Conceito

> **Aplica-se a:** Syhunt Hybrid 7.2

> **Observação**
> Este plano de testes foi elaborado para o **Syhunt Hybrid**, a edição completa. O Syhunt Community tem funcionalidades e verificações de vulnerabilidades restritas; por isso, alguns casos de teste não podem ser executados com ele e nem todas as vulnerabilidades serão detectadas.

Introdução

Este plano de testes verifica o funcionamento, a precisão e a correta operação de todas as partes principais do Syhunt Hybrid. Ele deve ser executado nas instalações do cliente, após a implantação e ativação do Syhunt Hybrid, e consiste em varreduras dinâmicas e estáticas executadas em diversas condições que simulam o uso real. Ao final, você estará familiarizado com os principais recursos do produto e com as opções de integração, e a Syhunt poderá coletar seu feedback sobre a usabilidade.

Casos de teste

Fase	Caso de teste
1. Auditoria dinâmica	D1: Varredura de vulnerabilidades em um site ativo
	D2: Login manual pelo navegador e varredura de um site restrito
	D3: Login automático e varredura de um site restrito
	D4: Apenas mapear um site (teste de rastreamento com execução de JavaScript)
	D5: Varredura dinâmica incremental
	D6: Varreduras dinâmicas simultâneas
2. Auditoria de código-fonte	S1: Varredura de arquivos de código-fonte locais
	S2: Varredura de repositórios Git remotos
3. Integração	N1: Iniciar uma varredura de código pela linha de comando
	N2: Iniciar uma varredura dinâmica pela linha de comando
	N3: Criar uma issue em um rastreador a partir de uma vulnerabilidade relatada
	N4: Gerar uma exportação compatível com o F5 BIG-IP ASM
4. Relatórios	R1: Gerar um relatório HTML padrão
	R2: Gerar um relatório HTML completo
	R3: Exportar os resultados da varredura em XML
	R4: Editar uma vulnerabilidade relatada
	R5: Gerar um relatório comparativo entre varreduras

Requisitos do ambiente

Os resultados deste plano foram obtidos no Windows 11 x64 com um Intel Core i9, 64 GB de RAM, SSD de 1 TB e acesso à Internet. Certifique-se de ter um SSD rápido e acesso à Internet, e de atender pelo menos aos requisitos mínimos de sistema.

Página 10 de 182

Syhunt Hybrid 7.2

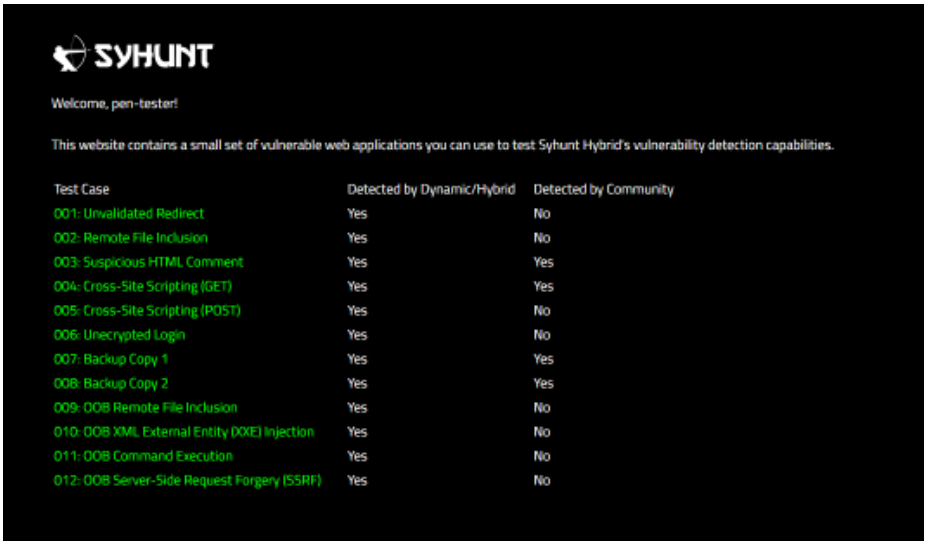
O plano pressupõe que o Syhunt Hybrid já esteja instalado na máquina Windows que executará os testes e que a interface web esteja habilitada. Caso não esteja:

1. Baixe o instalador pelo link fornecido pela Syhunt, clique duas vezes nele e siga o instalador (*Next → Next → Finish*). O Syhunt Hybrid é aberto quando você clica em **Finish**.
2. Habilite a interface web do Syhunt.
3. Na página inicial da interface web, informe a chave Pen-Tester Key recebida no e-mail que continha o link de download e clique em **Apply**. Uma mensagem de sucesso confirma que o Syhunt está pronto para os testes.

Prey Server

O Prey Server é um servidor web Apache/PHP portátil que inclui um conjunto de aplicações web propositalmente vulneráveis, para fins de demonstração.

1. Baixe-o em https://syhunt.fra1.cdn.digitaloceanspaces.com/tools/hybrid_xtras/syhunt-vulnphpserver.zip.
2. Descompacte-o em uma pasta de sua escolha.
3. Execute `PreyServer.exe`.
4. Execute `mysql\start.exe` para iniciar o servidor MySQL.
5. Abra http://127.0.0.1/syhunt/vulndemo no navegador. Você verá a página de boas-vindas:



Git for Windows

Os casos de teste relacionados ao Git exigem o Git for Windows. Instale-o com as configurações padrão e confirme que o comando `git` funciona no Prompt de Comando.

Tempo estimado das varreduras

- **Varreduras dinâmicas (fase 1)** levam cerca de **5 minutos** cada, nos alvos recomendados. A varredura incremental de D5 leva cerca de metade desse tempo.
- **Varreduras de código (fase 2)** levam cerca de **10 segundos** nos alvos recomendados. Para comparação, um projeto Java + JavaScript maior, com cerca de 200.000 linhas, leva em torno de 4 minutos, e uma nova varredura incremental do mesmo código leva cerca de metade disso. Ou seja, o Syhunt analisa aproximadamente 6 milhões de linhas de código por hora, e cerca de 12 milhões por hora em varreduras recorrentes dos mesmos alvos.

Fase 1: Auditoria dinâmica

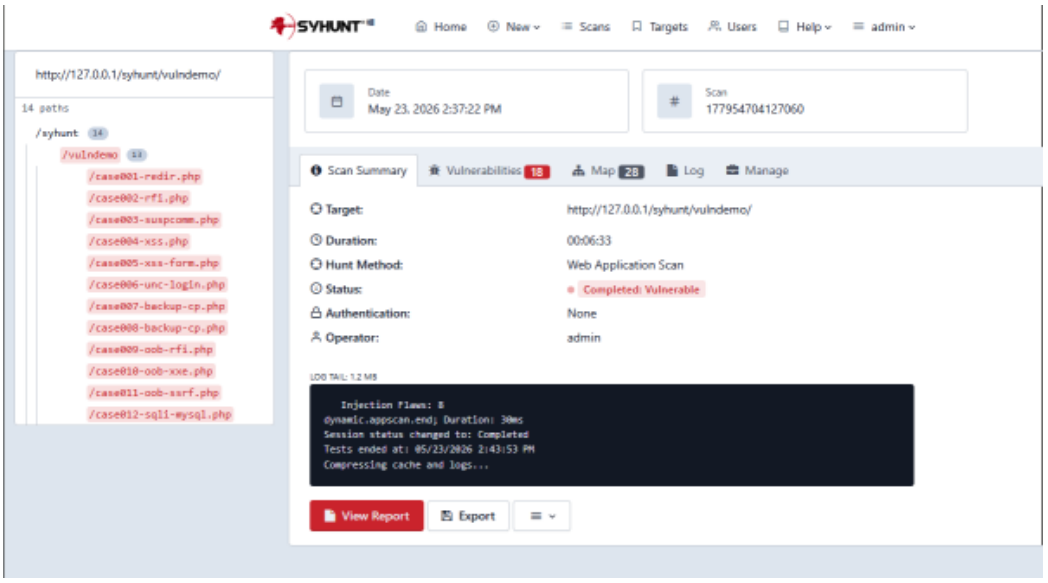
D1. Varredura de vulnerabilidades em um site ativo

Ação: fazer a varredura de http://127.0.0.1/syhunt/vulndemo em busca de vulnerabilidades.

Página 11 de 182

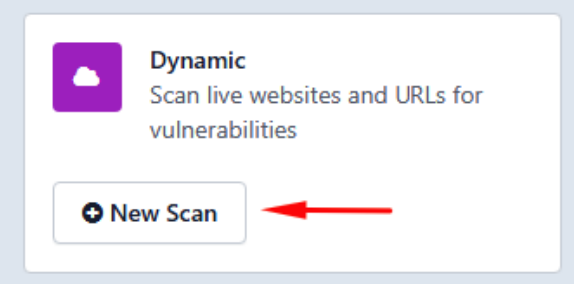
Syhunt Hybrid 7.2

Resultado esperado: ao final da varredura, a aba **Vulnerabilities** lista todos os casos de teste, de 001 a 012.

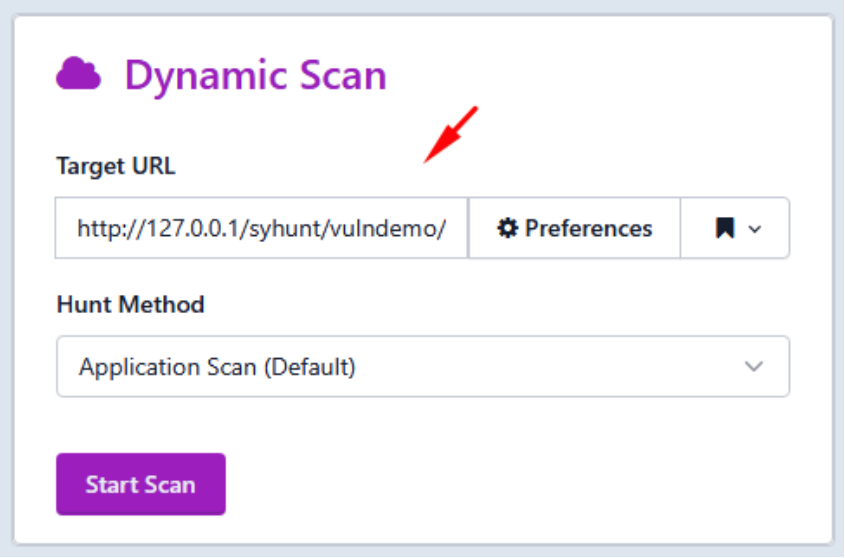


Passos:

1. Certifique-se de que o Prey Server está em execução (consulte Prey Server).
2. Abra a página inicial do Syhunt Hybrid, `https://127.0.0.1:8018/syhunt/`, e clique no botão **New Scan** do Dynamic.



3. Informe `http://127.0.0.1/syhunt/vulndemo` como URL alvo.



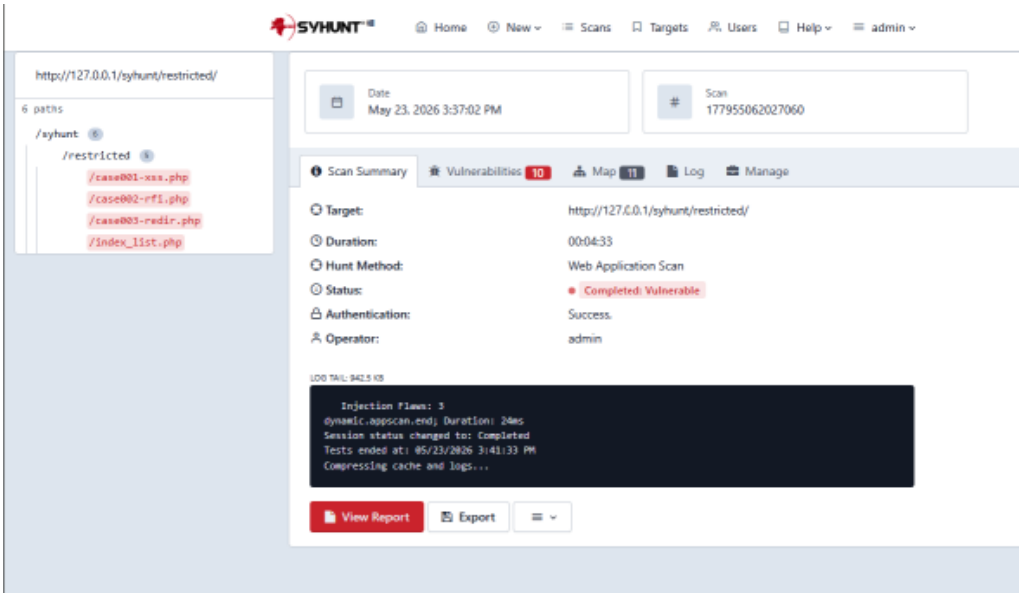
4. Selecione o método de varredura **Application Scan (Default)**, que procura todas as vulnerabilidades usando as configurações recomendadas.
5. Clique em **Start Scan**.

D2. Login manual pelo navegador e varredura de um site restrito

Ação: fazer a varredura de http://127.0.0.1/syhunt/restricted, que usa autenticação por formulário web.

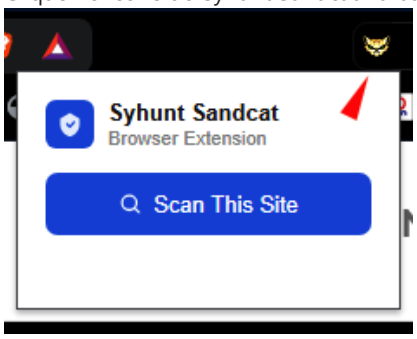
Página 12 de 182

Resultado esperado: ao final da varredura, a aba **Vulnerabilities** lista todos os casos de teste, de 001 a 003.



Passos:

1. Certifique-se de que o Prey Server está em execução (consulte Prey Server).
2. Certifique-se de que a extensão de navegador Syhunt Sandcat está habilitada.
3. No navegador, acesse http://127.0.0.1/syhunt/restricted.
4. Faça login com o usuário **test** e a senha **CUBPzjVy**.
5. Clique no ícone do Syhunt Sandcat na barra de ferramentas do navegador e depois em **Scan This Site**.



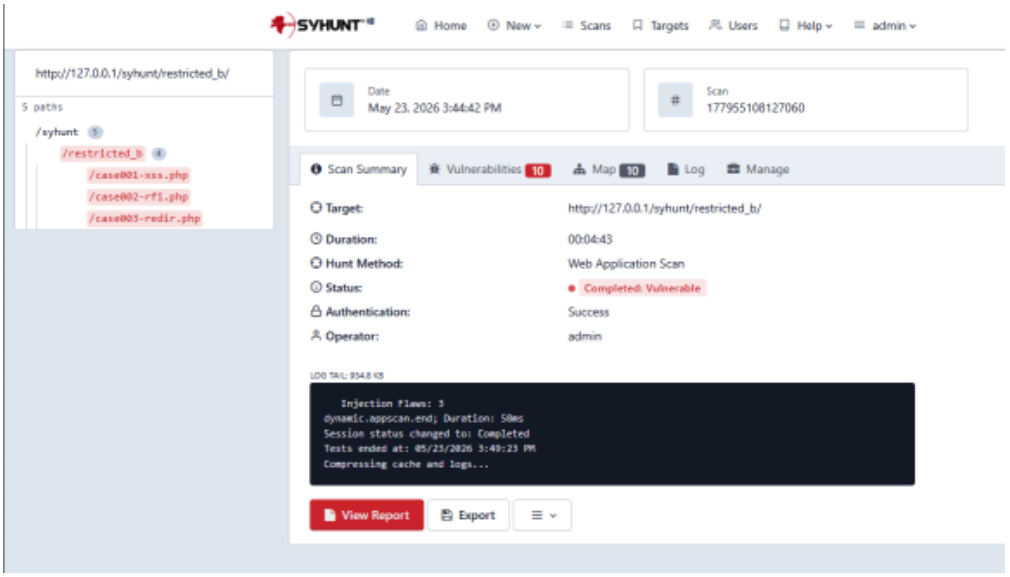
6. Aparece uma tela com uma captura do site e a mensagem verde **Session Ready**.
7. Selecione o método de varredura **Application Scan (Default)**.
8. Clique em **Start Scan**.

D3. Login automático e varredura de um site restrito

Ação: fazer a varredura de http://127.0.0.1/syhunt/restricted_b, que usa autenticação HTTP Basic.

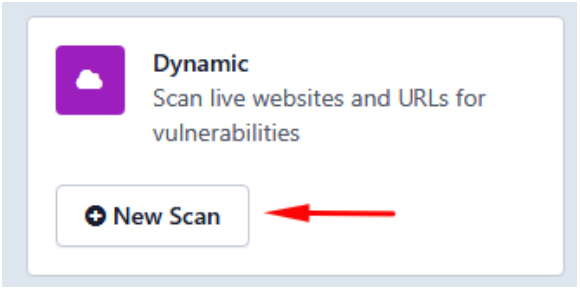
Resultado esperado: ao final da varredura, os resultados listam todos os casos de teste, de 001 a 003.

Syhunt Hybrid 7.2

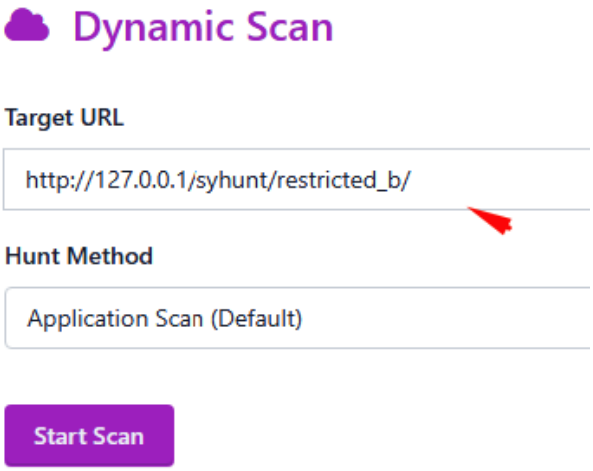


Passos:

1. Certifique-se de que o Prey Server está em execução (consulte Prey Server).
2. Abra `https://127.0.0.1:8018/syhunt/` e clique no botão **New Scan** do Dynamic.



3. Informe `http://127.0.0.1/syhunt/restricted_b` como URL alvo.

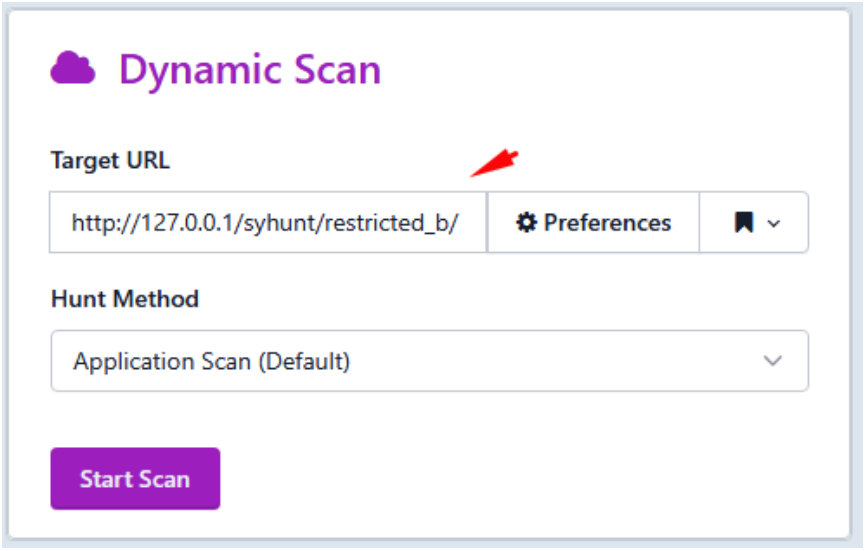


4. Selecione o método de varredura **Application Scan (Default)**.

Página 14 de 182

Syhunt Hybrid 7.2

5. Clique em **Preferences**.

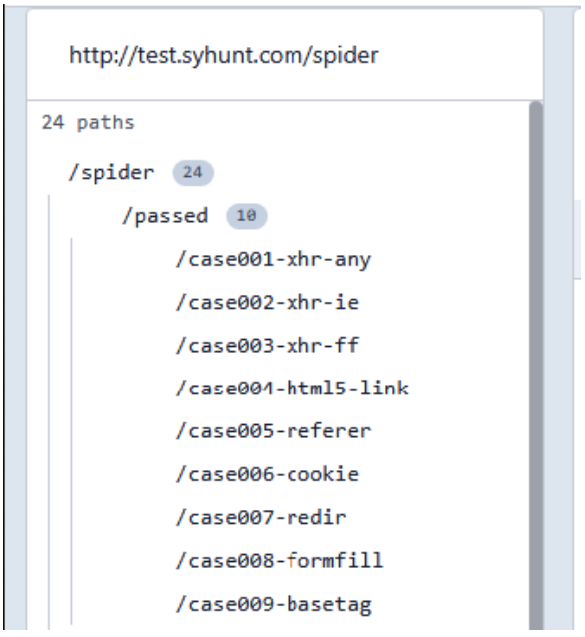


6. Na aba **Authentication**, defina **Authentication** como **Basic**.
7. Informe o usuário **test** e a senha **test** e clique em **Save**.
8. Clique em **Start Scan**.

D4. Apenas mapear um site (teste de rastreamento com execução de JavaScript)

Ação: rastrear http://test.syhunt.com/spider.

Resultado esperado: ao final da varredura, ao expandir a árvore do site na barra lateral esquerda, aparece a estrutura do site, incluindo todos os casos de teste de 001 a 009 no diretório `/passed/`.

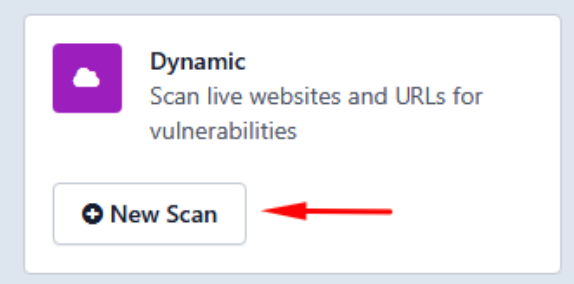


Passos:

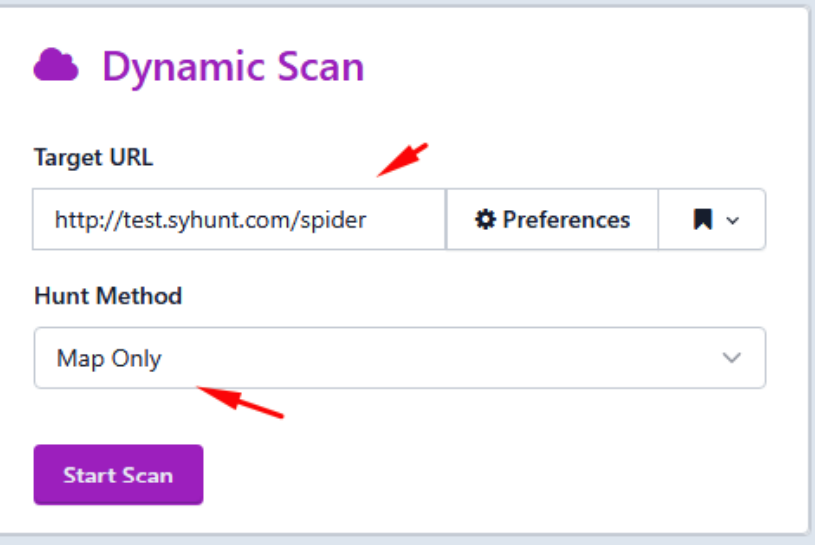
Página 15 de 182

Syhunt Hybrid 7.2

1. Abra `https://127.0.0.1:8018/syhunt/` e clique no botão **New Scan** do Dynamic.



2. Informe `http://test.syhunt.com/spider` como URL alvo.

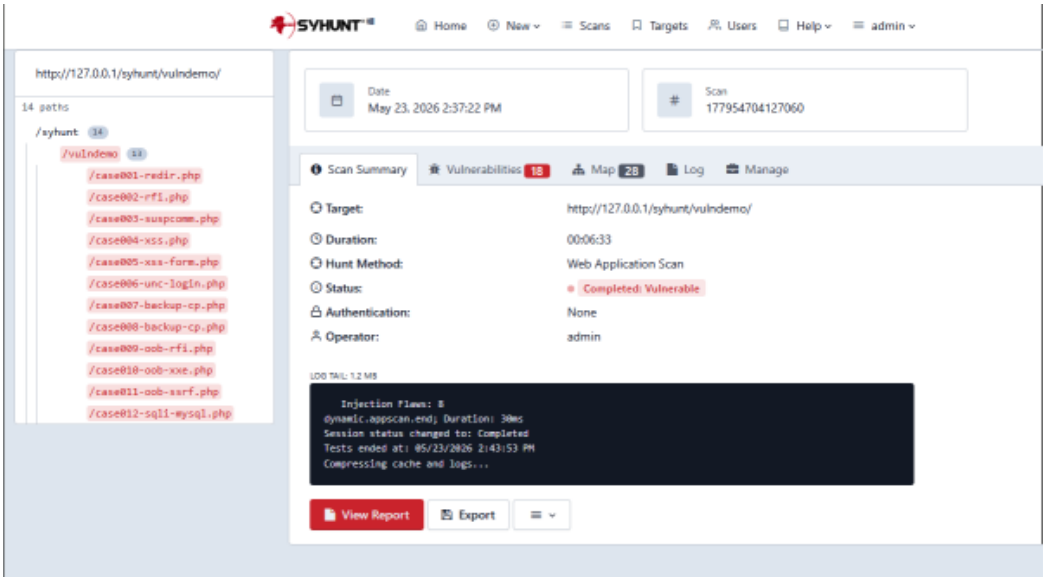


3. Selecione o método de varredura **Spider Only**.
4. Clique em **Start Scan**.

D5. Varredura dinâmica incremental

Ação: fazer duas varreduras de http://127.0.0.2/syhunt/vulndemo.

Resultado esperado: a primeira varredura leva cerca de 4 minutos; a segunda, cerca de metade disso (em torno de 2 minutos). Após cada varredura, os resultados listam todos os casos de teste, de 001 a 012.



Passos:

Página 16 de 182

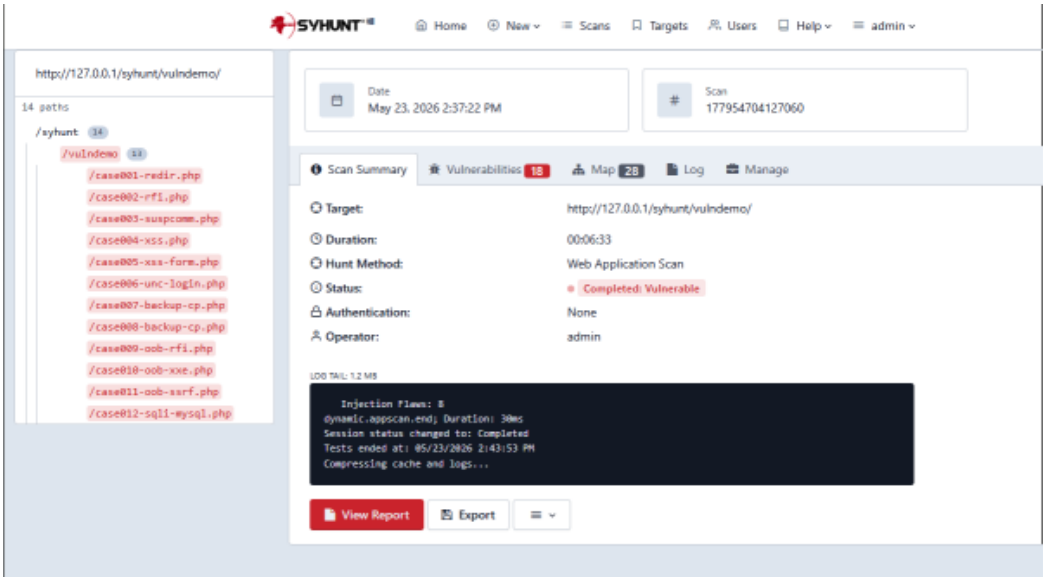
Syhunt Hybrid 7.2

1. Siga os passos de D1, mas use `127.0.0.2` em vez de `127.0.0.1` como alvo, pois a varredura incremental não fica habilitada para `127.0.0.1`.
2. Repita os mesmos passos para fazer uma segunda varredura de `127.0.0.2`.

D6. Varreduras dinâmicas simultâneas

Ação: executar duas varreduras simultâneas de http://127.0.0.1/syhunt/vulndemo.

Resultado esperado: ao final das varreduras, a aba **Vulnerabilities** de cada uma lista todos os casos de teste, de 001 a 012. A barra lateral direita mostra o status de cada varredura, marcado em vermelho.



Passos:

1. Siga os passos de D1.
2. Antes que a varredura termine, volte à tela inicial do Syhunt Hybrid e repita os passos 2 a 5 de D1 para iniciar uma segunda varredura do mesmo alvo.

Fase 2: Auditoria de código-fonte

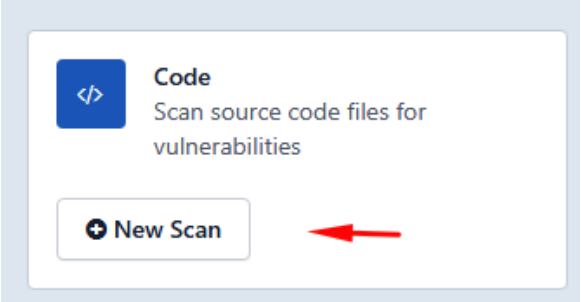
S1. Varredura de arquivos de código-fonte locais

Ação: iniciar uma varredura de código de um arquivo compactado local pela interface.

Resultado esperado: ao final da varredura, a aba **Vulnerabilities** lista todos os casos de teste, de 001 a 009.

Passos:

1. Baixe https://github.com/syhunt/vulnphp/archive/master.zip pelo navegador.
2. Abra a página inicial do Syhunt Hybrid e clique no botão **New Scan** do Code.



3. Na aba **File**, clique em **Choose File** e selecione o arquivo `master.zip` baixado.
4. Certifique-se de que o método de varredura **Code Scan (Default)** está selecionado.

Página 17 de 182

Syhunt Hybrid 7.2

5. Clique em **Start Scan**.

Você pode repetir este teste com amostras vulneráveis em outras linguagens (o número de casos de teste varia conforme o arquivo):

Linguagem	Arquivo
Java	https://github.com/syhunt/vulnjava-wavsep/archive/master.zip
Lua	https://github.com/syhunt/vulnlua/archive/master.zip
PHP	https://github.com/syhunt/vulnphp/archive/master.zip

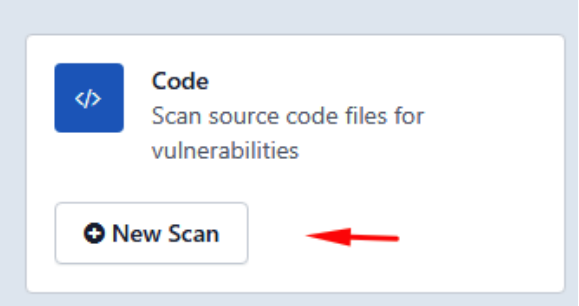
S2. Varredura de repositórios Git remotos

Ação: iniciar uma varredura de código de um repositório Git remoto.

Resultado esperado: ao final da varredura, os resultados listam todos os casos de teste, de 001 a 009.

Passos:

1. Abra a página inicial do Syhunt Hybrid e clique no botão **New Scan** do Code.



2. Informe `https://github.com/syhunt/vulnphp.git` como URL do projeto.
3. Certifique-se de que o método de varredura **Code Scan (Default)** está selecionado.
4. Clique em **Start Scan**.

Você pode repetir este teste com repositórios em outras linguagens (o número de casos de teste varia conforme o repositório):

Linguagem	Repositório
Java	`https://github.com/syhunt/vulnjava-wavsep.git`
Lua	`https://github.com/syhunt/vulnlua.git`
PHP	`https://github.com/syhunt/vulnphp.git`

Fase 3: Integração

N1. Iniciar uma varredura de código pela linha de comando

Ação: fazer a varredura de https://github.com/syhunt/vulnphp pela interface de linha de comando.

Resultado esperado: ao final da varredura, os resultados listam todos os casos de teste, de 001 a 009.

Passos:

1. Abra um Prompt de Comando na pasta de instalação do Syhunt Hybrid (consulte Linha de Comando).

2. Execute:

```
scancode https://github.com/syhunt/vulnphp.git
```


Página 18 de 182

Syhunt Hybrid 7.2

N2. Iniciar uma varredura dinâmica pela linha de comando

Ação: fazer a varredura de http://127.0.0.1/syhunt/vulndemo pela interface de linha de comando.

Resultado esperado: ao final da varredura, os resultados listam todos os casos de teste, de 001 a 012.

Passos:

1. Certifique-se de que o Prey Server está em execução (consulte Prey Server).
2. Abra um Prompt de Comando na pasta de instalação do Syhunt Hybrid (consulte Linha de Comando).
3. Execute:

```
scanurl 127.0.0.1/syhunt/vulndemo
```

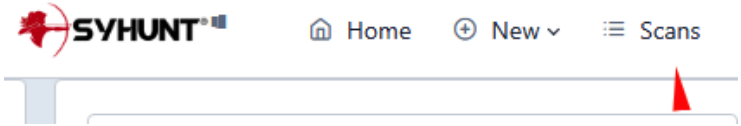
N3. Criar uma issue em um rastreador a partir de uma vulnerabilidade relatada

Ação: criar uma issue no GitHub a partir de uma vulnerabilidade relatada.

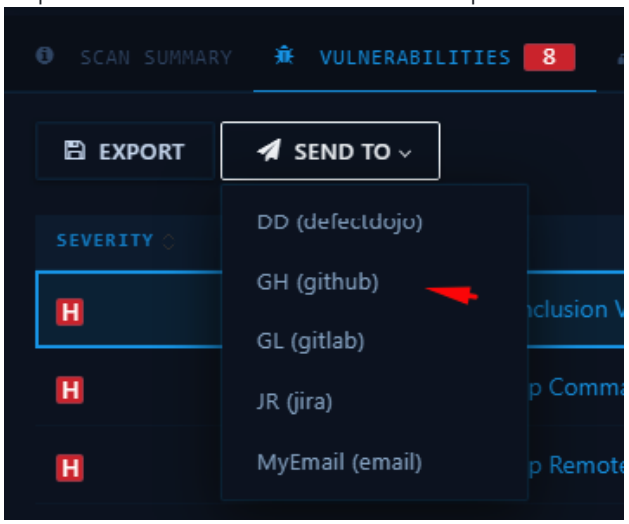
Resultado esperado: depois de adicionar um conector do GitHub e enviar uma vulnerabilidade, ela aparece como issue no seu repositório do GitHub.

Passos:

1. Crie uma conta no GitHub, se ainda não tiver, e crie um repositório de teste.
2. Adicione um conector do GitHub no Syhunt, conforme descrito em Rastreadores de Issues.
3. Abra **Scans** no menu.



4. Clique em uma varredura com status **Vulnerable** e vá para a aba **Vulnerabilities**.
5. Selecione a vulnerabilidade que deseja enviar ao GitHub.
6. Clique em **Send To** → ***nome do seu conector*** para enviá-la.



N4. Gerar uma exportação compatível com o F5 BIG-IP ASM

Ação: gerar um arquivo de exportação ASM para uma varredura.

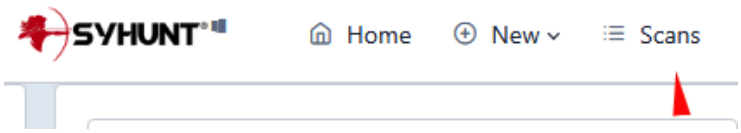
Resultado esperado: depois de clicar em **Download**, o navegador abre o arquivo XML compatível, que pode então ser importado no BIG-IP ASM.

Passos:

Página 19 de 182

Syhunt Hybrid 7.2

1. Abra **Scans** no menu.



2. Clique com o botão direito em uma varredura com status **Vulnerable** e escolha **Generate Report**.
3. Selecione o modelo de relatório **Complete**.
4. Defina **Format** como **XML ASM generic scanner format**.
5. Clique em **Generate** e depois em **Download**.

Fase 4: Relatórios

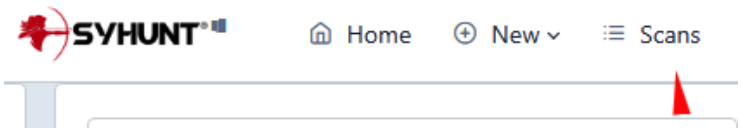
R1. Gerar um relatório HTML padrão

Ação: gerar um relatório de uma varredura, ordenando as vulnerabilidades pelo CVSS 3.

Resultado esperado: depois de clicar em **Generate**, o navegador permite baixar o relatório HTML.

Passos:

1. Abra **Scans** no menu.



2. Clique em uma varredura com status **Vulnerable**, vá para a aba **Vulnerabilities** e clique em **Export**.
3. Selecione o modelo de relatório **Standard**, se ainda não estiver selecionado.
4. Selecione o método de ordenação **CVSS**, se ainda não estiver selecionado.
5. Clique em **Generate** (canto inferior esquerdo da aba) e depois em **Download**.

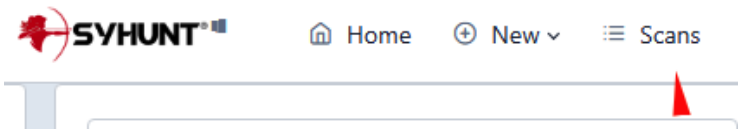
R2. Gerar um relatório HTML completo

Ação: gerar um relatório completo de uma varredura, ordenando as vulnerabilidades pelo CVSS 3.

Resultado esperado: depois de clicar em **Generate**, o navegador permite baixar o relatório HTML.

Passos:

1. Abra **Scans** no menu.



2. Clique em uma varredura com status **Vulnerable**, vá para a aba **Vulnerabilities** e clique em **Export**.
3. Selecione o modelo de relatório **Complete**.
4. Selecione o método de ordenação **CVSS**, se ainda não estiver selecionado.
5. Clique em **Generate** (canto inferior esquerdo da aba) e depois em **Download**.

Página 20 de 182

Syhunt Hybrid 7.2

R3. Exportar os resultados da varredura em XML

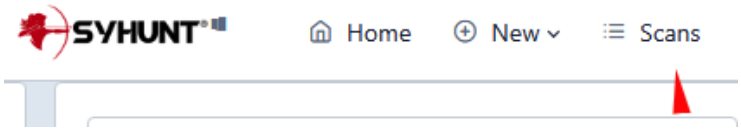
Ação: gerar um arquivo XML com os resultados de uma varredura.

Resultado esperado: depois de clicar em **Download**, o navegador permite baixar o arquivo XML de resultados.

```
</vulnerable_code>
▼<cvss>
  ▼<cvss3>
     <vector>AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H</vector>
     <score>9.8 CRITICAL</score>
     <base_score>9.8</base_score>
     <severity>critical</severity>
     <impact_score>5.9</impact_score>
     <exploitability_score>3.9</exploitability_score>
     <temporal_score>NA</temporal_score>
     <environmental_score>NA</environmental_score>
     <mod_impact_subscore>NA</mod_impact_subscore>
   </cvss3>
  ▼<cvss2>
     <vector>AV:N/AC:L/Au:N/C:C/I:C/A:C</vector>
     <score>10.0 HIGH</score>
     <base_score>10.0</base_score>
     <severity>high</severity>
     <impact_score>10.0</impact_score>
     <exploitability_score>10.0</exploitability_score>
```

Passos:

1. Abra **Scans** no menu.



2. Clique em uma varredura com status **Vulnerable**, vá para a aba **Vulnerabilities** e clique em **Export**.
3. Selecione o modelo de relatório **Complete**.
4. Defina o formato de saída como **XML**.
5. Clique em **Generate** e depois em **Download**.

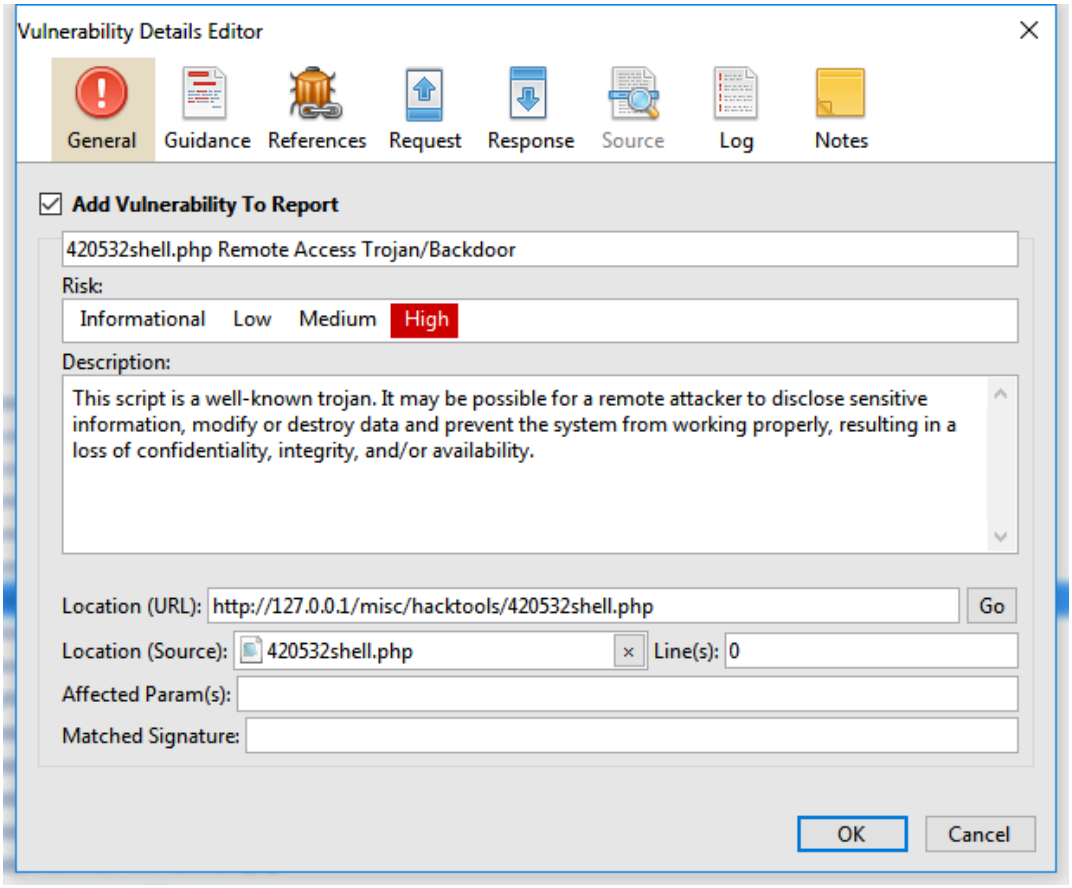
R4. Editar uma vulnerabilidade relatada

Ação: editar os detalhes de uma vulnerabilidade relatada.

Resultado esperado: a tela de edição da vulnerabilidade é aberta. Depois de salvar as alterações, todo relatório gerado inclui as informações editadas.

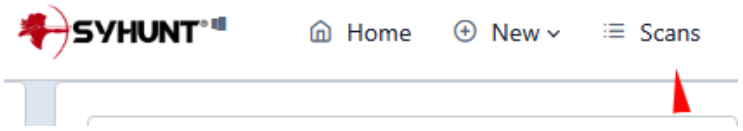
Página 21 de 182

Syhunt Hybrid 7.2



Passos:

1. Abra **Scans** no menu.

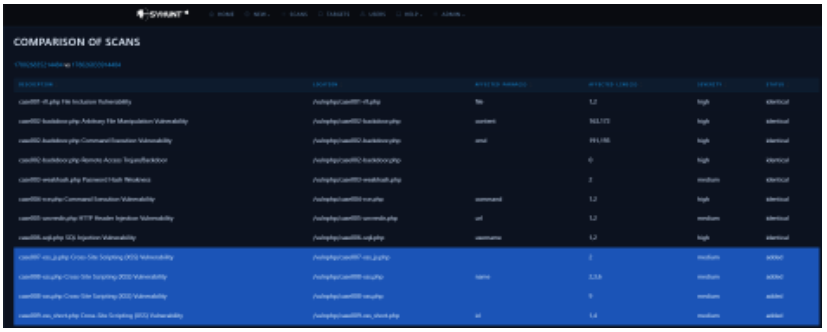


2. Clique em uma varredura com status **Vulnerable** e vá para a aba **Vulnerabilities**.
3. Clique em uma vulnerabilidade. A tela de detalhes é aberta.
4. Na aba **Notes**, adicione algumas observações.
5. Clique em **Save**.

R5. Gerar um relatório comparativo entre varreduras

Ação: gerar um relatório que compara duas varreduras.

Resultado esperado: o relatório gerado corresponde à tela de comparação e lista os casos de teste 007 a 009 como removidos.



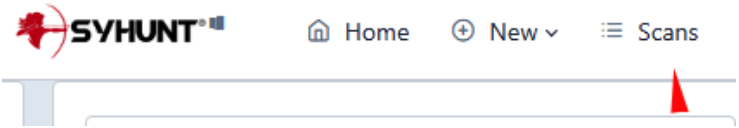
Passos:

1. Siga os passos de S1, mas extraia o arquivo para uma pasta e faça a varredura dessa pasta.

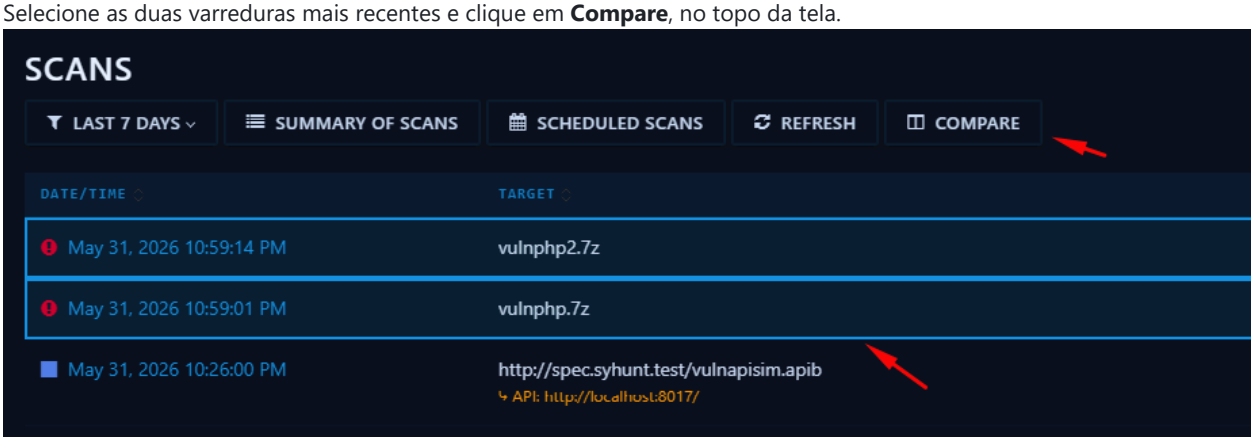
Página 22 de 182

Syhunt Hybrid 7.2

2. Ao final da varredura, exclua os casos de teste 007 a 009 dessa pasta.
3. Faça uma nova varredura da pasta.
4. Abra **Scans** no menu.



5. Selecione as duas varreduras mais recentes e clique em **Compare**, no topo da tela.



Página 23 de 182

Syhunt Hybrid 7.2

Perguntas Frequentes

> **Aplica-se a:** Syhunt Hybrid 7.2

Syhunt Dynamic

Quanto tempo o Syhunt Dynamic leva para executar todos os testes?

Depende de quantas páginas e aplicações o site contém e do método de varredura escolhido. Após o rastreamento (crawling), as verificações das aplicações web costumam ser a etapa mais longa, e sua duração aumenta conforme o tamanho do site.

Posso reabrir uma varredura anterior e gerar seus relatórios novamente?

Sim. Abra **Scans** no menu, clique na varredura na lista e depois em **View Report**, ou em **Export** para salvar os dados em um formato específico.

Existe uma lista dos testes realizados pela versão atual?

Sim. Abra **Help** → **Vulnerability Checks** ou consulte Verificações de Vulnerabilidades.

Algum dos testes pode derrubar o host alvo?

Algumas verificações de negação de serviço (DoS) podem derrubar o alvo. Você pode desativá-las escolhendo um método de varredura sem testes de DoS, como **Complete Scan, No DoS** (consulte Métodos de Varredura).

O Syhunt Dynamic tem problemas com firewalls pessoais?

Você precisará permitir que o Syhunt faça conexões de saída. Alguns firewalls de software também lidam mal com tráfego intenso, por isso não recomendamos executar um firewall pessoal na mesma máquina que o Syhunt.

Se a máquina que executa a varredura tiver um firewall com filtragem de saída, tente desativá-lo. Ocasionalmente, vimos firewalls bloquearem silenciosamente as conexões de um programa sem consultar o usuário.

O Syhunt pode fazer varredura das portas 21 (FTP) ou 23 (Telnet)?

Não. O Syhunt Dynamic não é um scanner de rede de uso geral; ele é especializado em aplicações web e APIs web.

Nomes de produtos

O que aconteceu com o Sandcat Platinum, o TrustSight e o Collapse?

O Sandcat Platinum foi renomeado para Sandcat Pro e, mais tarde, tornou-se o Syhunt Dynamic. As tecnologias TrustSight e Collapse foram incorporadas ao Syhunt Dynamic.

Página 24 de 182

Syhunt Hybrid 7.2

Instalação

Página 25 de 182

Syhunt Hybrid 7.2

Visão Geral da Instalação e Windows

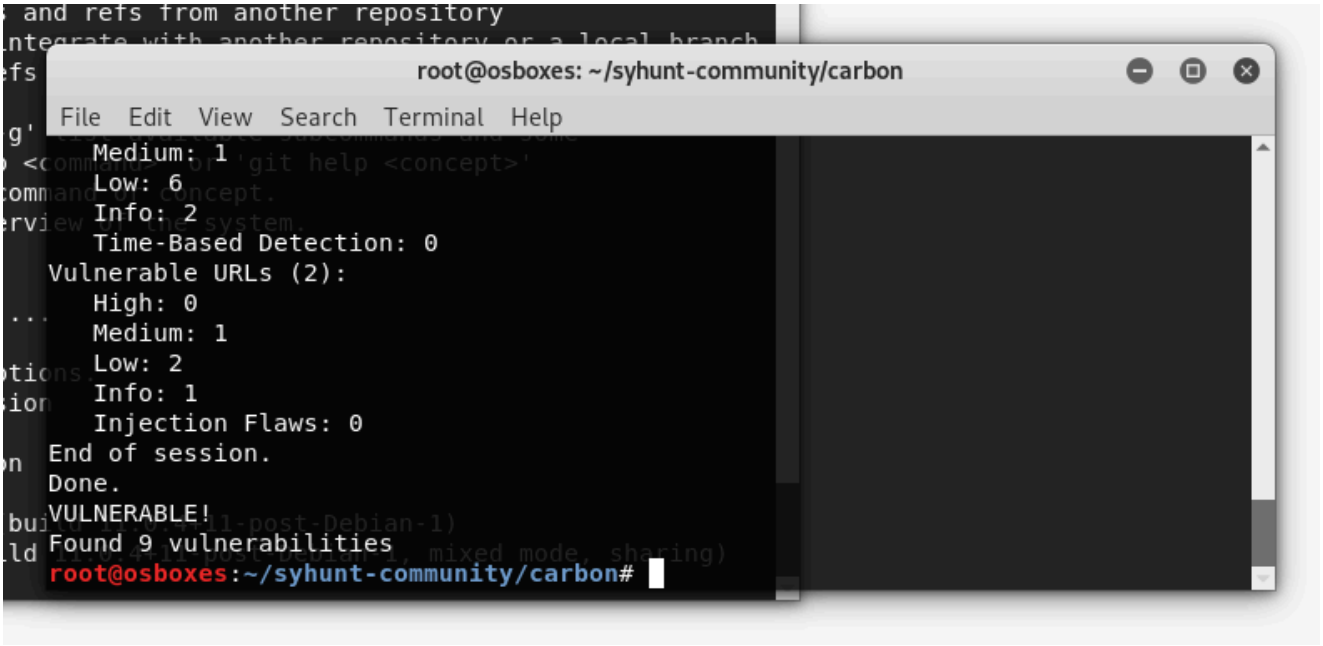
> **Aplica-se a:** Syhunt Hybrid 7.2

Conteúdo

- Introdução
- Edições Windows e Linux/macOS
- Instalação no Windows
- Instalação no macOS
- Instalação no Linux
- Instalação da extensão Syhunt Sandcat para navegador
- Requisitos de sistema
- Requisitos de conexão com a Internet
- Atualização do Syhunt
- Desinstalação do Syhunt

Introdução

O Syhunt Hybrid 7 é um scanner de segurança de aplicações multiplataforma de 64 bits. Ele roda no Windows, no macOS e em distribuições Linux como Ubuntu Desktop/Server, Debian e Fedora, praticamente sem esforço de configuração. O Syhunt Hybrid realiza testes de segurança de aplicações web e mobile (DAST e SAST) e se integra a diversos sistemas para integração e varredura contínuas.



Edições Windows e Linux/macOS

	Hybrid para Windows	**Hybrid para Linux/macOS**
Servidor da interface web	OpenResty para HTTPS e HTTP	NGINX para HTTPS, OpenResty como back-end
Rastreador AJAX avançado	✅ Motor Chromium	✅ Motor Firefox
Análise dinâmica aumentada (DAST + OAST)	✅	✅

Página 26 de 182

Syhunt Hybrid 7.2

	Hybrid para Windows	**Hybrid para Linux/macOS**
Análise de APIs (AAST + OAST)	✅	✅
Análise de código-fonte de aplicações web (SAST)	✅	✅
Análise de código-fonte de aplicações mobile (MAST)	✅	✅
Análise forense (FAST)	✅ (somente CLI)	✅ (somente CLI)
Recursos com IA	✅ Exemplos de código corrigido ✅ Pontuação de confiança por IA	✅ Exemplos de código corrigido ✅ Pontuação de confiança por IA
Autenticação avançada	✅ Recorder, Selenium	✅ Recorder, Selenium
Integrações	✅ Shell (PowerShell) ✅ Rastreadores de issues (GitHub, GitLab, Jira...) ✅ Ferramentas de CI/CD (GitLab, GitHub, Bitbucket, Jenkins) ✅ WAFs para virtual patching ✅ URLs de projetos Azure DevOps / TFS (não Git)	✅ Shell (Bash) ✅ Rastreadores de issues (GitHub, GitLab, Jira...) ✅ Ferramentas de CI/CD (GitLab, GitHub, Bitbucket, Jenkins) ✅ WAFs para virtual patching ✅ URLs de projetos Azure DevOps / TFS (não Git)
Agendador de varreduras	✅	✅
Sistemas operacionais suportados	**Oficialmente:** Windows 11 Windows 10 Windows Server 2016 a 2025	**Oficialmente:** macOS Sequoia 15.4.1 ou posterior (Apple Silicon) Ubuntu Server/Desktop 18.10 e posteriores Debian 9 e posteriores Fedora 32 e posteriores **Não oficialmente (testados com sucesso):** Outras distribuições Linux macOS Monterey
Disponibilidade	Disponível (binários nativos Win64)	Disponível (instalação baseada no Wine) Consulte sobre binários nativos para sua distribuição

Instalação no Windows

Clientes registrados: sigam os passos iniciais do Guia de Boas-Vindas para baixar, instalar e começar a usar a versão completa do Syhunt Hybrid. Usuários não registrados podem usar o Syhunt Community.

1. *(Opcional)* Se pretende fazer varreduras de repositórios Git, instale o Git for Windows.
2. Baixe o instalador do Syhunt Hybrid, `syhunt-hybrid-<versão>.exe`.
3. Clique duas vezes no arquivo baixado para executá-lo.
4. Leia atentamente o EULA antes de aceitá-lo.
5. Conclua o instalador (*Next* → *Next* → *Finish*). Ao clicar em **Finish**, o Syhunt é aberto e a tela do Launcher aparece.
6. Se necessário, habilite a interface web do Syhunt.

E agora? Leia o Início Rápido e os guias de integração.

Instalação no macOS

Consulte o guia de instalação para macOS.

Página 27 de 182

Syhunt Hybrid 7.2

Instalação no Linux

O Syhunt Hybrid para Linux é a edição *Carbon*: uma instalação baseada no Wine, compatível com a maioria dos desktops e servidores Linux de 64 bits. Consulte o guia de instalação para Linux para os passos específicos de cada distribuição.

Instalação da extensão Syhunt Sandcat para navegador

Desde a versão 7.2, o Syhunt Hybrid inclui uma extensão para o Edge, o Chrome, o Brave e outros navegadores baseados no Chromium. Ela permite o login manual e testes assistidos pelo navegador. O nome homenageia o Sandcat Browser, o navegador integrado do Syhunt 7.1 e versões anteriores, que oferecia login manual de forma semelhante.

1. No navegador, abra **Menu** → **Extensões** → **Gerenciar extensões**.
2. Ative o **Modo do desenvolvedor** (canto superior direito).
3. Clique em **Carregar sem compactação** e selecione a pasta da extensão Sandcat:
   - Windows: `C:\Program Files\Syhunt Hybrid\Extensions\SandcatExtension`
   - Linux: `/home/<usuário>/syhunt-hybrid/carbon/sandcat`
   - macOS: `/Applications/Syhunt Hybrid/carbon/sandcat`
4. O Syhunt Sandcat passa a aparecer na lista de extensões.
5. Abra novamente a página inicial da interface web. A mensagem **Extension detected and ready** confirma que tudo funciona.

Se você acessa a interface web por um endereço IP, domínio ou subdomínio diferente de `localhost`, configure esse endereço na extensão:

1. Clique em **Detalhes** na extensão e depois em **Opções da extensão**.
2. Selecione **Use custom Syhunt UI origin**.
3. Escolha o protocolo (`http` ou `https`).
4. Informe o host, *sem* o prefixo do protocolo (`http://` ou `https://`).
5. Informe a porta usada pela interface web.
6. Clique em **Save**.
7. Abra novamente a página inicial da interface web e verifique a mensagem **Extension detected and ready**.

Requisitos de sistema

Consulte as Especificações Técnicas.

Requisitos de conexão com a Internet

A máquina em que o Syhunt é executado deve conseguir fazer requisições HTTP(S) aos seguintes endereços:

Domínio	Portas	Finalidade
Seu(s) domínio(s) alvo	80, 443*	O domínio que hospeda a aplicação web ou o código que você quer verificar.
Quaisquer domínios de recursos, ou domínios específicos	80, 443	Opcional, mas recomendado: permite que o Syhunt analise arquivos JavaScript e outros recursos hospedados externamente durante o DAST ou o SAST.
`syhunt.fra1.cdn.digitaloceanspaces.com`	443	Obrigatório: recursos de instalação e atualização automática.
`www.syhunt.net`	80, 443	Obrigatório: recursos para relatórios HTML/PDF, notificações de atualização, atualização automática e mais.
`signal.syhunt.net`	80, 443	Obrigatório durante o DAST para realizar OAST (importante).
`api.openai.com`	443	Opcional: recursos com IA.
`fonts.googleapis.com`	80, 443	Obrigatório: fontes para relatórios HTML/PDF.

Página 28 de 182

Syhunt Hybrid 7.2

Domínio	Portas	Finalidade
`www.google.com`	443	Obrigatório: Google JSAPI para relatórios HTML/PDF.
`www.gstatic.com`	443	Obrigatório: Google JSAPI para relatórios HTML/PDF.

* Se o seu alvo usa uma porta não padrão (como 8080), ou se você se conecta ao Git por SSH ou outro protocolo, libere também essas portas.

> **Atenção**
> Se você usa um firewall pessoal, permita que o Syhunt faça conexões de saída com a Internet.

No Linux e no macOS, o instalador é um programa Java. Consulte Execução do instalador.

Atualização do Syhunt

As atualizações podem ser baixadas e instaladas a partir do site da Syhunt. O Syhunt Hybrid 6.8.4 e versões posteriores também anunciam novas versões no Launcher e nos relatórios.

No Linux e no macOS, execute `scanupdate` para verificar se há atualizações; se houver, o Syhunt pergunta se deve baixá-las e instalá-las. Execute `scanupdate auto` para instalar as atualizações disponíveis sem pedir confirmação.

Não é necessário desinstalar o Syhunt antes de instalar uma nova versão, exceto ao atualizar o Syhunt Community CLI no Windows.

Desinstalação do Syhunt

Windows: use o atalho de desinstalação no menu Iniciar (na pasta *Syhunt Hybrid* ou *Syhunt Community*) ou remova o Syhunt em **Painel de Controle** → **Programas e Recursos**.

Linux e macOS: vá até a pasta em que o Syhunt está instalado e execute:

```
java -jar Uninstall.jar
```


Página 29 de 182

Instalação no Linux

> **Aplica-se a:** Syhunt Hybrid 7.2

O Syhunt Hybrid roda no Ubuntu (Server ou Desktop) e na maioria das distribuições Linux de 64 bits praticamente sem esforço de configuração, e quase sem nenhum ajuste no Kali Linux. A edição para Linux (*Carbon*) é uma instalação baseada no Wine. Para comparar os recursos com a edição para Windows, consulte a Visão Geral da Instalação.

Visão geral

A instalação do Syhunt no Linux segue sempre o mesmo padrão:

1. Leia e aceite o EULA do Syhunt antes de executar qualquer um dos comandos abaixo.
2. Instale o **Java** (necessário para o instalador), se ainda não estiver instalado.
3. *(Opcional)* Instale o **Git** se pretende fazer varreduras de repositórios Git.
4. Instale o **Wine64**, seguindo o guia do Wine para sua distribuição.
5. Execute o instalador do Syhunt.

As seções abaixo trazem os comandos exatos para cada distribuição.

Distribuições: Kali · Parrot OS · Ubuntu · Debian · Fedora · CentOS / RHEL · AlmaLinux · Rocky Linux · Oracle Linux · Amazon Linux 2 · Amazon Linux 2023 · Arch / Manjaro · Linux Mint · MX Linux · openSUSE · elementary OS · Deepin · Solus

Kali Linux

> **Aplica-se a:** Kali Linux 2024.3 ou posterior

O Kali já inclui a maior parte das dependências do Syhunt, então você só precisa do Wine64 e do próprio Syhunt.

1. Instale o Wine64 seguindo o guia do Wine para Debian/Kali.
2. Execute o instalador do Syhunt.

Parrot OS

> **Aplica-se a:** Parrot OS 6.1 ou posterior

Assim como o Kali, o Parrot OS já inclui a maior parte das dependências do Syhunt.

1. Instale o Wine64 seguindo o guia do Wine para Debian.
2. Execute o instalador do Syhunt.

Ubuntu

> **Aplica-se a:** Ubuntu Desktop/Server 18.10 a 24.04 e posteriores; KDE neon 2020.03 ou posterior

1. Instale o Java:

```
sudo apt install openjdk-17-jre
```

Syhunt Hybrid 7.2

2. *(Opcional)* Instale o Git:

```
sudo apt install git
```

3. Instale o Wine64 seguindo o guia do Wine para Ubuntu.
4. Execute o instalador do Syhunt.

Debian

> **Aplica-se a:** Debian 9, 10 ou 12 (com interface gráfica ou mínima)

1. Instale o Wine64 seguindo o guia do Wine para Debian.
2. Instale o Java (no Debian 9, provavelmente já está instalado):

```
sudo apt install default-jre
```

3. *(Opcional)* Instale o Git:

```
sudo apt install git
```

4. Execute o instalador do Syhunt.

Fedora

> **Aplica-se a:** Fedora 32, 39, 40 e 41

1. Aplique todas as atualizações pendentes e reinicie.
2. Instale o Java:

```
sudo yum install java
```

3. *(Opcional)* Instale o Git:

```
sudo yum install git
```

4. Instale o Wine64 seguindo o guia do Wine para Fedora.
5. Execute o instalador do Syhunt.

> **Dica**
> Se o OpenJDK 17 estiver instalado e o instalador falhar com um erro como `Exception in thread "main" java.lang.UnsatisfiedLinkError: Can't load library: /usr/lib/jvm/java-17-openjdk-…/lib/libawt_xawt.so`, instale os pacotes do OpenJDK que faltam e execute o instalador novamente:
>
> ```
> sudo dnf install java-17-openjdk java-17-openjdk-devel java-17-openjdk-headless
> ```


Página 31 de 182

Syhunt Hybrid 7.2

CentOS e Red Hat Enterprise Linux

> **Aplica-se a:** CentOS 7 ou 8 (Everything ou Minimal); Red Hat Enterprise Linux

1. Instale o Wine64 seguindo o guia do Wine para CentOS.
2. Instale o Java:

```
sudo yum -y update
sudo yum install java    # ou: sudo yum install java-1.8.0-openjdk
```

3. *(Opcional; no Minimal, necessário para varreduras de repositórios Git)* Instale o Git:

```
sudo yum install git
```

4. *(Instalações Minimal)* Instale o wget:

```
sudo yum install wget
```

5. Execute o instalador do Syhunt.

AlmaLinux, Rocky Linux e Oracle Linux

> **Aplica-se a:** AlmaLinux 9, Rocky Linux 9, Oracle Linux 9

1. Instale o Wine64 seguindo o guia do Wine para sua distribuição: AlmaLinux, Rocky Linux ou Oracle Linux.
2. Instale o Java:

```
sudo yum -y update
sudo yum install java    # ou: sudo yum install java-1.8.0-openjdk
```

3. *(Opcional)* Instale o Git:

```
sudo yum install git
```

4. *(AlmaLinux e Rocky Linux)* Instale o wget:

```
sudo yum install wget
```

5. Permita que processos não confinados executem arquivos que exigem a permissão `execmod` do SELinux. Isso resolve um conflito com o Wine; enfraquece levemente o SELinux, mas não o desativa.

```
sudo setsebool -P selinuxuser_execmod on
```

6. Execute o instalador do Syhunt.

Amazon Linux 2

1. Instale o Wine64 seguindo o guia do Wine para Amazon Linux 2.
2. Instale o Java:

```
sudo yum -y update
```


Página 32 de 182

Syhunt Hybrid 7.2

```
sudo yum install java
```

3. *(Opcional)* Instale o Git:

```
sudo yum install git
```

4. Execute o instalador do Syhunt.

Amazon Linux 2023

1. Instale o Wine64 seguindo o guia do Wine para Amazon Linux 2023.
2. Instale o Java:

```
sudo yum -y update
sudo yum install java
```

3. *(Opcional)* Instale o Git:

```
sudo yum install git
```

4. Permita a permissão `execmod` do SELinux, como explicado na seção do AlmaLinux:

```
sudo setsebool -P selinuxuser_execmod on
```

5. Execute o instalador do Syhunt.

Arch Linux e Manjaro

> **Aplica-se a:** Arch Linux 2019 ou posterior, Manjaro 19

1. Atualize o banco de dados do pacman:

```
sudo pacman -Syy
```

2. Instale o Java 8:

```
sudo pacman -S jre8-openjdk
```

3. *(Opcional)* Instale o Git:

```
sudo pacman -S git
```

4. Instale o Wine64:
   - **Arch Linux:** siga o guia do Wine para Arch Linux.
   - **Manjaro:** execute `sudo pacman -S wine` e depois `wine64 ipconfig`. O Wine é iniciado e se configura sozinho; pressione **I** se ele pedir para instalar componentes ausentes.
5. Execute o instalador do Syhunt.

Linux Mint

> **Aplica-se a:** Linux Mint 18, 19 e 20

Página 33 de 182

Syhunt Hybrid 7.2

1. O Java já vem instalado no Linux Mint.
2. *(Opcional)* Instale o Git:

```
sudo apt install git
```

3. Instale o Wine64 seguindo o guia do Wine para Linux Mint.
4. Execute o instalador do Syhunt.

MX Linux

> **Aplica-se a:** MX Linux 19.1 ou posterior

1. O Java e o Git já vêm instalados no MX Linux 19.
2. Instale o Wine64 e execute-o uma vez para que ele se configure:

```
sudo apt install wine
wine64 ipconfig
```

3. Execute o instalador do Syhunt.

openSUSE

> **Aplica-se a:** openSUSE Leap 15, Leap 42 ou posterior, Tumbleweed

1. *(Opcional)* Instale o Git:

```
zypper in git        # ou, se não for encontrado: zypper in git-core
```

2. Instale o Wine64 seguindo o guia do Wine para openSUSE.
3. Execute o instalador do Syhunt.

elementary OS

> **Aplica-se a:** elementary OS 5.0 e 5.1

- **elementary OS 5.1:** siga os passos do Ubuntu. Pode ser necessário reiniciar após instalar o Wine64.
- **elementary OS 5.0:** leia primeiro este aviso. Se ainda quiser prosseguir, faça isso por sua conta e risco.

Deepin

> **Aplica-se a:** Deepin 15.9

1. Instale o Java:

```
sudo apt install default-jre
```

2. *(Opcional)* Instale o Git:

Página 34 de 182

Syhunt Hybrid 7.2

```
sudo apt install git
```

3. Instale o Wine64 e execute-o uma vez para que ele se configure:

```
sudo apt install wine
wine64 ipconfig
```

4. Execute o instalador do Syhunt.

Solus

> **Aplica-se a:** Solus 4.1

1. Instale o Java (o comando `java` ficará em `/usr/lib64/openjdk-8/bin/java` ou `/usr/lib64/openjdk-11/bin/java`):

```
sudo eopkg install openjdk-8    # ou: sudo eopkg install openjdk-11
```

2. *(Opcional)* Instale o Git:

```
sudo eopkg install git
```

3. Instale o Wine64 seguindo o guia do Wine para Solus.
4. Execute o instalador do Syhunt.

Execução do instalador (Linux e macOS)

1. Baixe o instalador do Syhunt Hybrid com um navegador, `wget` ou `curl`.

2. Execute o instalador:

Edição	Comando
Hybrid	`java -jar syhunt-hybrid-<versão>.jar`
Hybrid em servidor sem interface gráfica	`java -jar syhunt-hybrid-<versão>.jar -console`
Community	`java -jar <arquivo>`

   No Linux, se o `binfmt-support` estiver instalado (`sudo apt install binfmt-support`), você pode tornar o arquivo executável (`chmod a+rx <arquivo>.jar`) e executá-lo diretamente, por exemplo `./syhunt-hybrid-<versão>.jar` (acrescente `-console` em servidores sem interface gráfica).

3. Leia atentamente o EULA antes de aceitá-lo e conclua o instalador (*Next* → *Next* → *Finish*). Por padrão, o Syhunt é instalado em:

   - Linux: `/home/<usuário>/syhunt-hybrid` ou `/home/<usuário>/syhunt-community`
   - macOS: `/Applications/Syhunt Hybrid/` ou `/Applications/Syhunt Community/`

4. Habilite a interface web do Syhunt.

Modo console

Se o instalador for executado em modo console:

1. Leia o EULA e digite **1** para aceitá-lo (ou **2** para recusá-lo; o Syhunt não pode ser instalado sem aceitá-lo).
2. Pressione **Enter** para usar o caminho de instalação padrão.

Página 35 de 182

Syhunt Hybrid 7.2

3. Digite **1** para continuar.
4. Digite **Y** para instalar o Syhunt Core.
5. Digite **1** para continuar. A mensagem **Console installation done** confirma a instalação.

E agora? Leia o Início Rápido e os guias de integração.

Página 36 de 182

Syhunt Hybrid 7.2

Instalação no macOS

> **Aplica-se a:** Syhunt Hybrid 7.2 no macOS Sequoia e versões anteriores

Conteúdo

- Introdução
- Instalação no macOS (Apple Silicon ou Intel)
- Parallels Desktop: Debian ARM64
- Parallels Desktop: Ubuntu ou Kali Linux ARM64
- Parallels Desktop: habilitando o Rosetta
- Parallels Desktop: instalando o Wine 8.0

Introdução

O Syhunt Hybrid 7.2 é compatível com o macOS Sequoia e versões anteriores. A edição para macOS é a edição *Carbon*: ela é instalada sobre o Wine e funciona tanto em Macs com Apple Silicon (por meio do Rosetta) quanto em Macs com Intel.

Se preferir virtualização, o Syhunt Hybrid 7.1.6 também roda em Linux ARM64 (Ubuntu, Debian ou Kali Linux) dentro do Parallels Desktop no macOS. Isso exige uma licença **Pro** do Parallels; a licença Standard não é compatível.

Instalação no macOS (Apple Silicon ou Intel)

1. Leia e aceite o EULA do Syhunt antes de executar qualquer um dos comandos abaixo.
2. Baixe o Java para macOS (`jre-<versão>-macosx-x64.dmg`) e instale-o, caso ainda não o tenha.
3. Instale o Wine64, caso ainda não o tenha:
   - Apple Silicon (M1, M2, M3 ou M4)
   - Intel
4. Instale o wget:

   ```
   brew install wget
   ```

5. Baixe o instalador do Syhunt (arquivo `.jar`), clique no ícone com a tecla Control pressionada e escolha **Abrir**. Leia atentamente o EULA antes de aceitá-lo e conclua o instalador (*Next* → *Next* → *Finish*). As ferramentas de linha de comando do Syhunt ficam então prontas para uso. Para mais opções, consulte Execução do instalador.
6. Habilite a interface web do Syhunt.

> **Observação**
> Se, após uma atualização do macOS, a varredura de um repositório Git falhar com **invalid active developer path**, execute `xcode-select --install` para reparar o comando `git` .

Parallels Desktop: Debian ARM64

> **Aplica-se a:** Debian 12.6 ARM64 em Macs com Apple Silicon usando o Parallels Desktop

1. Instale o Java:

   ```
   sudo apt install default-jre
   ```


Página 37 de 182

Syhunt Hybrid 7.2

2. *(Opcional)* Instale o Git, se pretende fazer varreduras de repositórios Git:

```
sudo apt install git
```

3. Habilite e instale o Rosetta (consulte Habilitando o Rosetta).
4. Instale a versão AMD64 do Wine64. O Debian 12.6 ARM64 instala a versão 8.0 por padrão:

```
sudo apt install wine64:amd64
```

5. Execute `wine ipconfig` uma vez para que o Wine se configure.
6. Instale o Syhunt: baixe o instalador `.jar` e siga Execução do instalador.
7. Habilite a interface web do Syhunt.

Parallels Desktop: Ubuntu ou Kali Linux ARM64

> **Aplica-se a:** Ubuntu 24.04 ARM64 ou Kali Linux 2024.2 ARM64 em Macs com Apple Silicon usando o Parallels Desktop

1. Leia e aceite o EULA do Syhunt.
2. Crie uma máquina virtual dedicada ao Syhunt.
3. Atualize as listas de pacotes:

```
sudo apt-get update
```

4. Instale o Java com o SVE desativado:

```
# Adicione uma vez ao perfil do shell
echo 'export JAVA_TOOL_OPTIONS="$JAVA_TOOL_OPTIONS -XX:UseSVE=0"' | sudo tee /etc/profile.d/java-
nosve.sh
# Recarregue o ambiente de login (ou abra um novo terminal)
source /etc/profile.d/java-nosve.sh
# Instale o OpenJDK 17 e confirme a versão
sudo apt install openjdk-17-jre
java --version
```

5. Habilite e instale o Rosetta (consulte Habilitando o Rosetta). Pule este passo se estiver usando a imagem gratuita **Ubuntu with Rosetta** do Parallels.
6. O pacote padrão do Wine é a versão 9, que não funciona bem com o Rosetta. Primeiro habilite o Wine 8.0 (consulte Instalando o Wine 8.0) e depois instale-o:

```
sudo apt install libwine:amd64=8.0~repack-4 wine64:amd64=8.0~repack-4
```

   Se o comando `wine` não estiver disponível depois disso, execute `sudo apt install wine`. Confirme a versão com `wine --version`.
7. Execute `wine ipconfig` uma vez para que o Wine se configure.
8. Instale o Syhunt: baixe o instalador `.jar` e siga Execução do instalador.
9. Habilite a interface web do Syhunt.

Parallels Desktop: habilitando o Rosetta

> **Aplica-se a:** Ubuntu 24.04 ARM64 ou Kali Linux 2024.2 ARM64 em Macs com Apple Silicon usando o Parallels Desktop

1. Quando a instalação da máquina virtual terminar, desligue-a. Nas configurações da VM, vá em **Hardware → CPU & Memory**, ative **Use Rosetta to run x86_64 binaries** e ligue a VM novamente.

Página 38 de 182

Syhunt Hybrid 7.2

2. Instale o suporte ao Rosetta:

```
wget https://kb.parallels.com/Attachments/kcs-193687/rosetta_x86_sources.sh
chmod +x rosetta_x86_sources.sh
sudo ./rosetta_x86_sources.sh enable    # exibe "AMD64 architecture enabled"
sudo apt update
```

Parallels Desktop: instalando o Wine 8.0

> **Aplica-se a:** Ubuntu 24.04 ARM64 ou Kali Linux 2024.2 ARM64 em Macs com Apple Silicon usando o Parallels Desktop

> **Atenção**
> Use este procedimento apenas em ambientes de teste. Ele instala pacotes do Debian 12 (*Bookworm*) no Ubuntu 24.04 ou no Kali Linux 2024.2. Atualmente funciona, mas misturar pacotes de distribuições diferentes não é uma solução estável a longo prazo. Estamos trabalhando em um método mais limpo e atualizaremos este guia quando ele estiver disponível.

1. Adicione o repositório do Debian 12, fixado para ser usado apenas pelo Wine:

```
echo 'deb http://deb.debian.org/debian bookworm main contrib non-free non-free-firmware' |
  sudo tee /etc/apt/sources.list.d/debian-bookworm.list

sudo tee /etc/apt/preferences.d/99-wine8 <<'EOF'
Package: wine* libwine* fonts-wine*
Pin: release n=bookworm
Pin-Priority: 990

Package: *
Pin: release n=bookworm
Pin-Priority: 100
EOF
```

2. Adicione a chave de assinatura do repositório Bookworm:

```
sudo mkdir -p /etc/apt/keyrings &&
curl -fsSL https://ftp-master.debian.org/keys/archive-key-12.asc | gpg --dearmor | sudo tee
/etc/apt/keyrings/debian-bookworm.gpg > /dev/null &&
echo "deb [signed-by=/etc/apt/keyrings/debian-bookworm.gpg] http://deb.debian.org/debian bookworm main"
| sudo tee /etc/apt/sources.list.d/debian-bookworm.list &&
sudo apt update
```

3. Adicione os repositórios do Ubuntu para pacotes AMD64:

```
sudo tee /etc/apt/sources.list.d/ubuntu-amd64.list > /dev/null <<EOF
deb [arch=amd64] http://archive.ubuntu.com/ubuntu noble main restricted universe multiverse
deb [arch=amd64] http://archive.ubuntu.com/ubuntu noble-updates main restricted universe multiverse
deb [arch=amd64] http://archive.ubuntu.com/ubuntu noble-security main restricted universe multiverse
deb [arch=amd64] http://archive.ubuntu.com/ubuntu noble-backports main restricted universe multiverse
EOF
```

4. Atualize o APT e confirme que o Wine 8.0 está disponível. A saída deve incluir `8.0~repack-4`:

```
sudo apt update
apt-cache madison wine | grep 8.0
```

5. Continue a partir do passo 6 das instruções para Ubuntu/Kali.

Página 39 de 182

Syhunt Hybrid 7.2

Interface Web

> **Aplica-se a:** Syhunt Hybrid 7.2

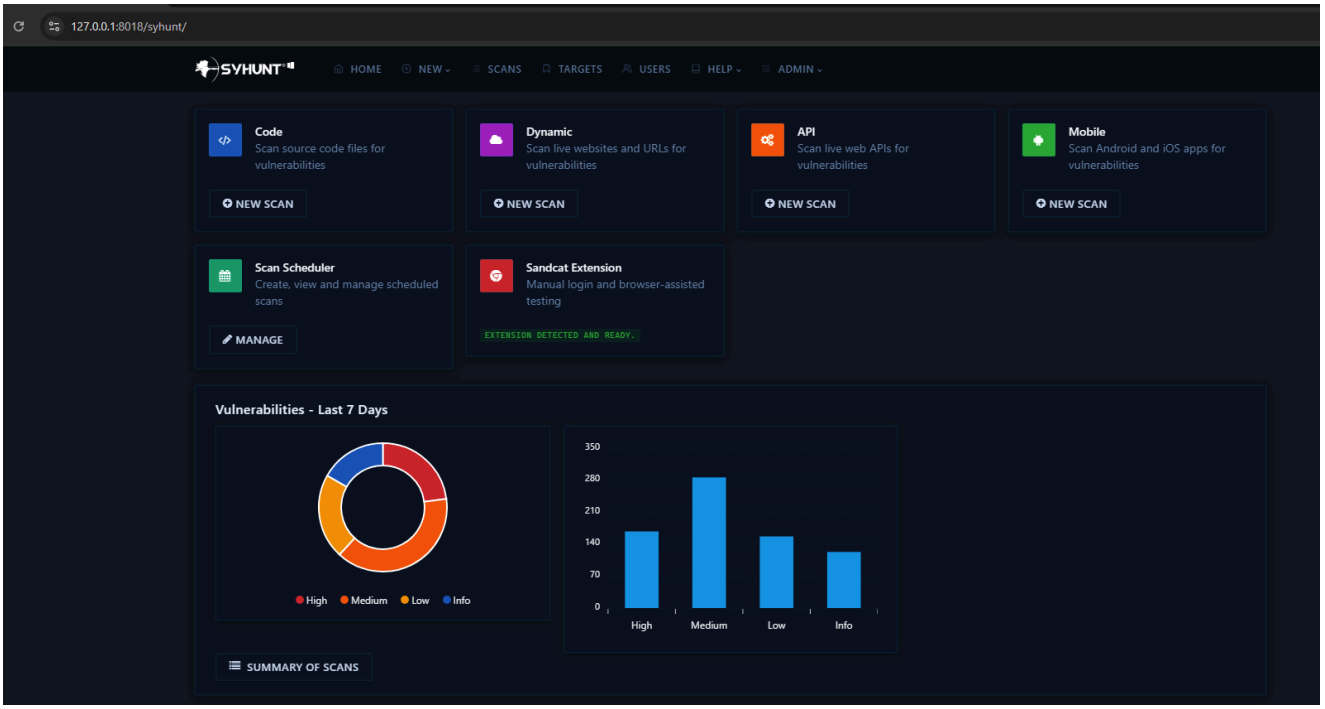
Conteúdo

- Introdução
- Habilitando a interface web
- Criando contas de usuário adicionais
- Alterando o idioma padrão da interface
- Política de senhas
- Perfis de usuário
- Dicas

Introdução

Desde a versão 7.1.6, o Syhunt Hybrid inclui uma interface web que pode ser habilitada no Windows, no Linux e no macOS. Por ela, você pode iniciar varreduras DAST, SAST e MAST; ver os resultados de todas as varreduras, iniciadas pela interface, pela CLI, pela API REST ou pelo agendador; gerar relatórios e exportações; gerenciar usuários, permissões e sessões ativas; executar comandos de console; e muito mais. A interface web e os relatórios estão disponíveis em oito idiomas: inglês, alemão, espanhol, francês, italiano, japonês, coreano e português.

A interface web roda sobre o OpenResty (NGINX + Lua). Ela armazena as senhas com hash bcrypt de alto custo, gera IDs de sessão de forma segura e protege o login contra ataques de força bruta.



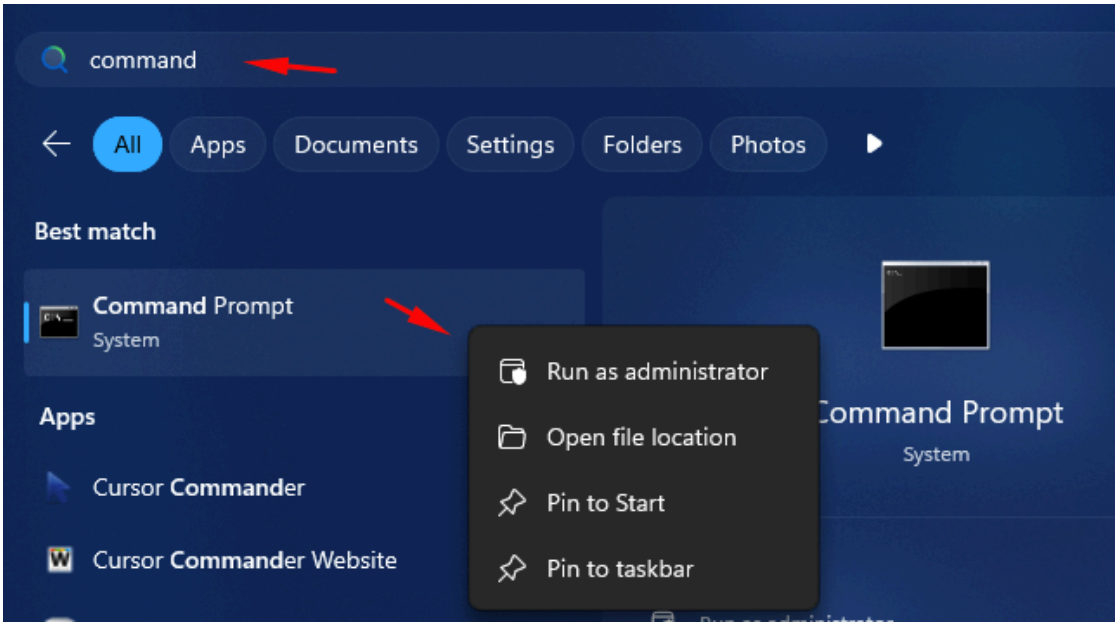
Habilitando a interface web

1. Abra um terminal ou Prompt de Comando na pasta da CLI do Syhunt (o local padrão depende do sistema operacional):
   - **Linux:** `cd ${HOME}/syhunt-hybrid/carbon/`
   - **macOS:** `cd "/Applications/Syhunt Hybrid/carbon"`

Página 40 de 182

Syhunt Hybrid 7.2

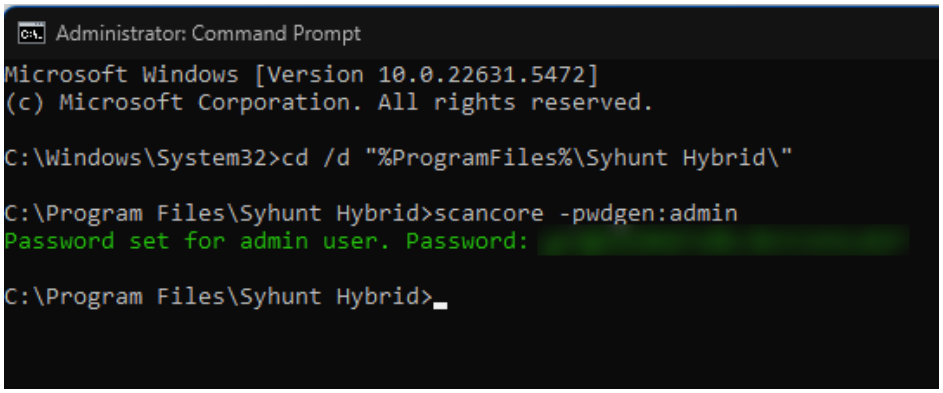
- **Windows:** abra o menu Iniciar e digite `command`. Clique com o botão direito em **Prompt de Comando** e escolha **Executar como administrador**,



depois execute `cd /d "%ProgramFiles%\Syhunt Hybrid\"`.

2. Configure a conta de administrador. Execute o comando a seguir, que gera e define a senha do administrador, e guarde a senha exibida; você precisará dela para fazer login:

```
scancore -pwdgen:admin
```

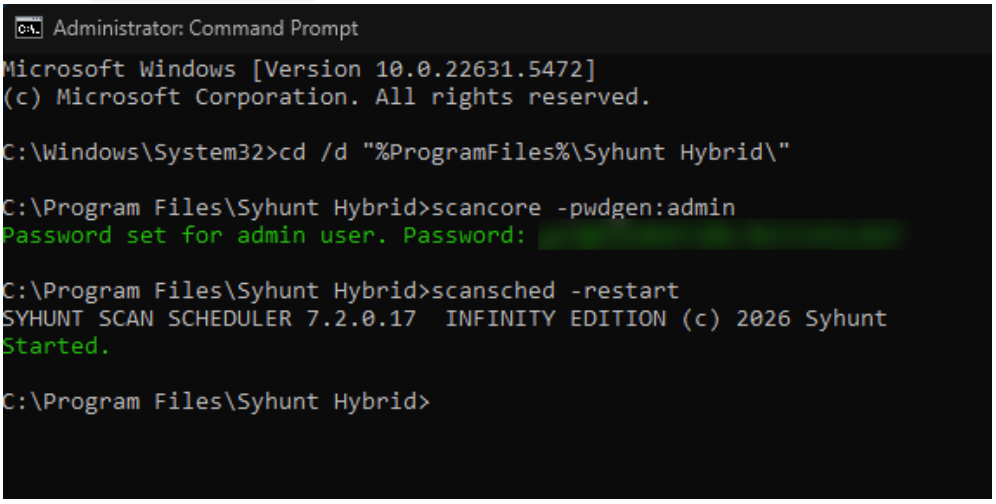


3. Inicie o servidor:
   - **Linux ou macOS:** execute `syservicereg` e depois `webui_utils --sslenable`. O serviço da interface web é instalado e iniciado.

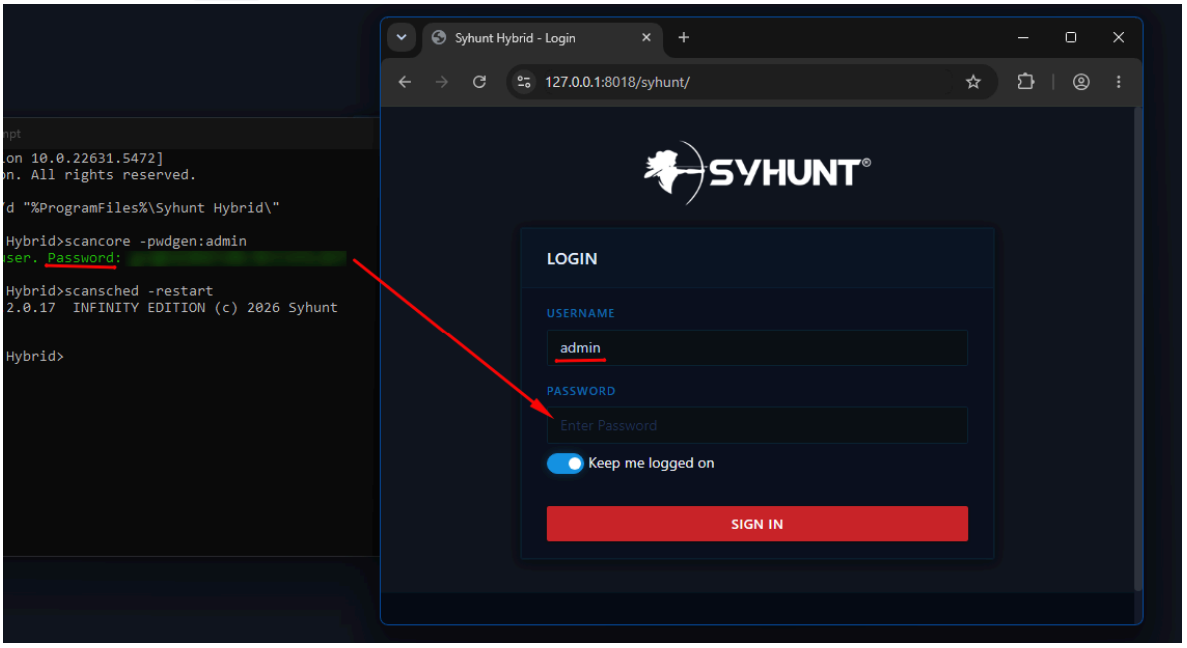
Página 41 de 182

Syhunt Hybrid 7.2

- **Windows:** reinicie o Windows (a interface web é iniciada junto com o sistema) ou, para evitar a reinicialização, execute `scansched -restart` como administrador.



4. Abra a interface web:
   1. No navegador, acesse `https://127.0.0.1:8018/syhunt/`.
   2. Informe o usuário `admin` e a senha gerada.



   3. Clique em **Sign In**.

Observações

- No **Windows**, depois que a senha do administrador é definida, a interface web é iniciada sempre que o serviço Hybrid Scheduler é iniciado, de modo que continua em execução mesmo quando o usuário atual faz logoff. No Linux, o serviço Syhunt Web UI faz o mesmo.
- No **AlmaLinux**, se o serviço da interface web não iniciar, talvez seja necessário desativar o SELinux permanentemente. Após reiniciar, o serviço deve funcionar normalmente:

```
sudo sed -i 's/^SELINUX=.*/SELINUX=disabled/' /etc/selinux/config && sudo reboot
```

Criando contas de usuário adicionais

Para criar outra conta, execute:

Página 42 de 182

Syhunt Hybrid 7.2

```
scancore -pwdgen:nomedousuario
```

Alterando o idioma padrão da interface

Para alterar o idioma da interface web, execute o comando a seguir com um dos códigos de idioma disponíveis:

```
scancore -langset:CODIGO
```

Por exemplo, `scancore -langset:PT` para português.

Política de senhas

Para reforçar a segurança e apoiar a conformidade, o Syhunt 7.1 introduziu uma política de senhas rigorosa para a interface web:

Requisito	Contas de administrador	Demais contas
Comprimento mínimo	26 caracteres	16 caracteres
Reutilização de senhas	Não é possível reutilizar as últimas 24 senhas	Não é possível reutilizar as últimas 10 senhas

Além disso, toda senha deve:

- **não** incluir o nome de usuário nem qualquer parte do nome completo do usuário;
- conter pelo menos uma letra maiúscula (A–Z), uma letra minúscula (a–z), um dígito (0–9) e um caractere não alfanumérico (como `@`, `#`, `$`, `%`);
- **não** conter três ou mais caracteres idênticos consecutivos (como `AAA`) nem três caracteres consecutivos em ordem crescente ou decrescente (como `123`, `321`, `ABC`, `CBA`).

Essas regras são verificadas sempre que uma senha é definida. Após uma atualização, se uma senha existente violar alguma delas, você precisará redefini-la no próximo login.

A forma mais fácil de criar uma senha válida é `scancore -pwdgen:<usuário>`, que gera uma senha com segurança, define-a e a exibe. Para digitar a senha você mesmo, use `-pwdset`.

Perfis de usuário

Conectado como administrador, você pode atribuir a qualquer usuário o perfil Operator, Restricted Operator ou Viewer:

Perfil	Pode
Administrator	Iniciar varreduras e ver todos os resultados, editar permissões e preferências de usuários e executar comandos de console. Atualmente, só é permitida uma conta de administrador.
Operator	Iniciar varreduras e ver os resultados de todas as varreduras.
Restricted Operator	Iniciar varreduras, mas ver apenas os resultados das suas próprias varreduras.
Viewer	Ver os resultados de todas as varreduras, mas não iniciar varreduras.

Permissões por perfil

Permissão	Admin	Operator	Restricted Operator	Viewer
Alertas administrativos na tela inicial	✅			
Excluir varreduras, alvos, usuários e conectores	✅			

Página 43 de 182

Syhunt Hybrid 7.2

Permissão	Admin	Operator	Restricted Operator	Viewer
Controle de acesso de usuários				
Acessar a tela Users	✅			
Criar usuários	✅			
Alterar o perfil de um usuário	✅			
Acessar perfis de usuário	✅	Apenas o próprio	Apenas o próprio	Apenas o próprio
Invalidar sessões de usuário	✅	Apenas as próprias	Apenas as próprias	Apenas as próprias
Alterar a senha de um usuário	✅	Apenas a própria	Apenas a própria	Apenas a própria
Varreduras e alvos				
Iniciar varreduras	✅	✅	✅	
Acessar a tela Targets e as preferências	✅	✅	✅	✅ (somente leitura)
Acessar informações das varreduras (logs, comparação, achados, resumo, mapa, relatórios, exportações, depuração, lista)	✅	✅	Apenas das próprias varreduras	✅
Acessar informações de ações (incluindo pré-visualização)	✅	✅	Apenas das próprias varreduras	✅
Telas administrativas				
Acessar a lista de varreduras agendadas	✅	✅	✅	✅
Acessar os detalhes de cada varredura agendada	✅			
Acessar a tela Connectors	✅			
Acessar o console de comandos	✅			
Acessar as preferências e listas globais (incluindo exportação)	✅			
Acessar o log do agendador (se habilitado)	✅	✅		
Importação				
Importar especificações de API	✅	✅		
Inserir ou atualizar a licença do produto	✅			

Dicas

Horário do agendador incorreto

Se o horário exibido na tela Scan Scheduler não corresponder ao horário do sistema, defina o fuso horário do sistema (substitua `America/Sao_Paulo` pelo seu):

- **macOS:** `sudo /usr/sbin/systemsetup -settimezone "America/Sao_Paulo"`
- **Linux:** `sudo timedatectl set-timezone America/Sao_Paulo`

Varredura de repositórios privados

Antes de fazer a varredura de um repositório Git privado pela interface web, configure as credenciais do repositório ou as chaves SSH e teste a varredura com o comando `scancode`.

Página 44 de 182

Syhunt Hybrid 7.2

Limitações

Conexões diretas a repositórios Azure DevOps (TFS) que não sejam Git não são suportadas no Linux nem no macOS.

Página 45 de 182

Syhunt Hybrid 7.2

Instalando o Wine

O Syhunt Hybrid para Linux e macOS (a edição *Carbon*) roda sobre o Wine64. Esta página explica como instalar o Wine64 em cada sistema suportado. Depois de instalar o Wine, volte ao guia de Linux ou de macOS para instalar o Syhunt.

> **Importante**
> Leia e aceite o EULA do Syhunt antes de executar qualquer um dos comandos abaixo.

Sistemas: macOS (Apple Silicon) · macOS (Intel) · Ubuntu · Debian / Kali · Fedora · CentOS · AlmaLinux · Rocky Linux · Oracle Linux · Amazon Linux 2 · Amazon Linux 2023 · Arch Linux · Linux Mint · openSUSE · elementary OS · Solus

> **Dica**
> Em todos os sistemas, o último passo é executar `wine64 ipconfig` uma vez. O Wine é iniciado pela primeira vez e se configura sozinho; se ele pedir para instalar componentes ausentes (como o Wine Mono ou o Wine Gecko), pressione **I** para instalá-los. Ao final, aparece uma mensagem como `wine: configuration in '/home/<usuário>/.wine' has been updated`.

macOS (Apple Silicon)

> **Aplica-se a:** macOS Sequoia em Apple Silicon (M1, M2, M3, M4)

1. Instale o Homebrew, caso ainda não o tenha. Abra o **Terminal** (em `/Applications/Utilities/`) e execute:

   ```
   /bin/bash -c "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/master/install.sh)"
   ```

   Informe sua senha de administrador quando solicitado. Ao final, o Homebrew exibe instruções para configurar o `~/.zprofile`; siga-as.
2. Instale o XQuartz (talvez seja preciso informar a senha de administrador novamente):

   ```
   brew install --cask xquartz
   ```

3. Instale o Rosetta:

   ```
   softwareupdate --install-rosetta --agree-to-license
   ```

4. Instale o Wine:

   ```
   brew install --cask --no-quarantine wine-stable
   ```

   Se o seu Homebrew não aceitar `--no-quarantine`, execute estes comandos:

   ```
   brew install --cask wine-stable
   xattr -dr com.apple.quarantine /Applications/Wine\ Stable.app
   ```

5. Após a mensagem `wine-stable was successfully installed!`, execute `wine64 ipconfig`.

Página 46 de 182

Syhunt Hybrid 7.2

macOS (Intel)

> **Aplica-se a:** macOS Big Sur e Monterey em Intel

1. Instale o Homebrew, caso ainda não o tenha. Abra o **Terminal** e execute:
   - macOS Catalina e posteriores:

     ```
     /bin/bash -c "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/master/install.sh)"
     ```

   - macOS High Sierra, Sierra, El Capitan e anteriores:

     ```
     /usr/bin/ruby -e "$(curl -fsSL
     https://raw.githubusercontent.com/Homebrew/install/master/install)"
     ```

   Informe sua senha de administrador quando solicitado e siga as instruções do Homebrew para configurar o `~/.zprofile`.

2. Instale o XQuartz:

   ```
   brew install --cask xquartz
   ```

3. Instale o Wine (consulte a seção do Apple Silicon para a alternativa caso `--no-quarantine` não seja aceito):

   ```
   brew install --cask --no-quarantine wine-stable
   ```

4. Execute `wine64 ipconfig`.

Ubuntu

> **Aplica-se a:** Ubuntu 18.10 a 24.04 e posteriores

```
sudo apt install wine-stable
wine ipconfig
```

O Syhunt é de 64 bits e não precisa do Wine de 32 bits. Se quiser instalá-lo mesmo assim:

```
sudo dpkg --add-architecture i386 && sudo apt-get update
sudo apt-get install wine32
```

Debian e Kali Linux

> **Aplica-se a:** Debian 9, 10 ou 12 (com interface gráfica ou mínima); Kali Linux 2024.3

```
sudo apt-get update
sudo apt install wine
wine64 ipconfig
```


Página 47 de 182

Fedora

> **Aplica-se a:** Fedora 32, 39 e 40

1. Verifique se há atualizações:

   ```
   sudo dnf check-update
   ```

2. Execute `wine64`. Quando for perguntado se deseja instalar o `wine-core`, digite **yes** e depois **Y** para confirmar as alterações.

   Se o download dos pacotes do Wine falhar, aplique todas as atualizações, reinicie e tente novamente:

   ```
   sudo dnf update
   sudo reboot
   sudo dnf install wine
   ```

3. Execute `wine64 ipconfig` e pressione **I** para instalar o Wine Mono e o Wine Gecko quando solicitado. O Wine pode exibir mensagens de depuração em segundo plano; elas podem ser ignoradas.

CentOS

> **Aplica-se a:** CentOS 7.7, 7.8, 8.0 e 8.2 (Everything ou Minimal)

1. Atualize o sistema:

   ```
   sudo yum -y update
   ```

   Se isso falhar com **No URLs in mirrorlist** (o CentOS 8 chegou ao fim de vida), aponte os repositórios para o vault do CentOS. O DNF voltará a funcionar, mas você não receberá mais atualizações (detalhes):

   ```
   sed -i 's/mirrorlist/#mirrorlist/g' /etc/yum.repos.d/CentOS-Linux-*
   sed -i 's|#baseurl=http://mirror.centos.org|baseurl=http://vault.centos.org|g' /etc/yum.repos.d/CentOS-Linux-*
   sudo yum -y update
   ```

2. Instale o EPEL e o Wine:

   ```
   sudo yum install epel-release
   sudo yum install wine
   ```

   Se o Wine64 não puder ser instalado porque falta o SDL2, instale primeiro o SDL2 e depois execute `sudo yum install wine` novamente:

   - Habilite o repositório que fornece o SDL2:

     ```
     sudo dnf config-manager --set-enabled PowerTools
     ```

   - CentOS 8.2.3 e posteriores:

     ```
     sudo dnf --enablerepo=PowerTools install SDL2
     ```

   - CentOS 8.4 e posteriores:

```
sudo dnf -y update
sudo dnf groupinstall 'Development Tools' -y
sudo dnf config-manager --set-enabled powertools
sudo dnf -y install SDL2-devel
```

3. Execute `wine64 ipconfig`.

AlmaLinux e Rocky Linux

> **Aplica-se a:** AlmaLinux 9, Rocky Linux 9

```
sudo yum -y update
sudo dnf install yum-utils
sudo yum install epel-release
sudo dnf config-manager --set-enabled crb
sudo dnf install mingw64-zlib
sudo yum install wine
wine64 ipconfig
```

Oracle Linux

> **Aplica-se a:** Oracle Linux 9

```
sudo yum -y update
sudo yum install epel-release
sudo yum install SDL2-devel
sudo yum install wine
```

Se o Wine64 não puder ser instalado porque nenhum pacote fornece `mingw64-zlib` e `mesa-libOSMesa`, habilite o repositório CodeReady Builder, instale o `mingw64-zlib` e instale o Wine novamente:

```
sudo dnf config-manager --set-enabled ol9_codeready_builder
sudo dnf install mingw64-zlib
sudo yum install wine
```

Por fim, execute `wine64 ipconfig`.

Amazon Linux 2

```
sudo yum update -y
sudo yum groupinstall "Development Tools" -y
sudo amazon-linux-extras install epel -y
```

Instale duas dependências importantes do Wine a partir de pacotes compatíveis do Oracle Linux e, em seguida, o próprio Wine:

```
sudo yum install -y https://yum.oracle.com/repo/OracleLinux/OL8/appstream/x86_64/getPackage/spirv-tools-libs-
2023.1-2.el8.x86_64.rpm
sudo yum install -y https://yum.oracle.com/repo/OracleLinux/OL7/9/base/x86_64/getPackage/isdn4k-utils-3.2-
99.el7.x86_64.rpm
sudo yum install -y wine
wine64 ipconfig
```

Syhunt Hybrid 7.2

Amazon Linux 2023

1. Adicione os repositórios de que o Wine precisa, usando o script auxiliar da Syhunt:

```
sudo yum groupinstall "Development Tools" -y
wget https://syhunt.fra1.cdn.digitaloceanspaces.com/tools/linux/al2023-createrepos.sh
chmod +x al2023-createrepos.sh
./al2023-createrepos.sh
```

2. Instale o Wine e execute-o uma vez:

```
sudo yum update -y
sudo yum install wine -y
wine64 ipconfig
```

3. Para evitar conflitos, remova os repositórios do Oracle Linux 9 adicionados pelo script, que não são mais necessários:

```
sudo rm /etc/yum.repos.d/epel.repo
sudo rm /etc/yum.repos.d/appstream.repo
sudo rm /etc/yum.repos.d/crb.repo
```

Arch Linux

> **Aplica-se a:** Arch Linux 2019

1. Habilite o suporte a multilib no pacman: edite `/etc/pacman.conf` (por exemplo, `sudo vi /etc/pacman.conf`), remova o comentário destas linhas e salve o arquivo:

```
[multilib]
Include = /etc/pacman.d/mirrorlist
```

2. *(Opcional)* Atualize as chaves do pacman:

```
sudo pacman-key --init
sudo pacman-key --populate archlinux
sudo pacman-key --keyserver keyserver.ubuntu.com --refresh-keys
sudo pacman -S archlinux-keyring
```

3. Atualize o banco de dados de pacotes e instale o Wine:

```
sudo pacman -Syy
sudo pacman -S wine
```

4. Execute `wine64 ipconfig` e pressione **I** para instalar o Wine Mono e o Wine Gecko quando solicitado.

Linux Mint

> **Aplica-se a:** Linux Mint 18, 19 e 20

Linux Mint 20

```
sudo apt-get update
sudo apt install wine-stable
wine64 ipconfig
```


Página 50 de 182

Linux Mint 18 e 19

1. Habilite a arquitetura de 32 bits (exigida pelos pacotes, embora o próprio Syhunt seja de 64 bits) e instale o Wine:

```
sudo dpkg --add-architecture i386
sudo apt install wine-stable
```

2. Se a instalação falhar por problemas de dependências (como aconteceu em nossos testes), instale o Wine a partir do repositório do WineHQ:

```
wget -nc https://dl.winehq.org/wine-builds/winehq.key; sudo apt-key add winehq.key
sudo apt-add-repository 'deb https://dl.winehq.org/wine-builds/ubuntu/ bionic main'
sudo add-apt-repository ppa:cybermax-dexter/sdl2-backport
sudo apt update && sudo apt install --install-recommends winehq-stable
```

   Use `bionic` para o Linux Mint 19.x ou `xenial` para o Linux Mint 18.x. O PPA fornece a biblioteca `libfaudio0`, que é necessária.
3. Execute `wine64 ipconfig`.

openSUSE

> **Aplica-se a:** openSUSE Leap 15, Leap 42 ou posterior, Tumbleweed

1. Adicione o repositório do Wine para a sua versão do openSUSE. O padrão da URL é `https://download.opensuse.org/repositories/Emulators:/Wine/openSUSE_<nome>_<versão>/Emulators:Wine.repo`. Por exemplo:

```
# Leap 15.1
sudo zypper ar -r
https://download.opensuse.org/repositories/Emulators:/Wine/openSUSE_Leap_15.1/Emulators:Wine.repo
# Tumbleweed
sudo zypper ar -r
https://download.opensuse.org/repositories/Emulators:/Wine/openSUSE_Tumbleweed/Emulators:Wine.repo
```

2. Atualize o cache do zypper. Quando for perguntado se deseja confiar na chave do novo repositório, digite **a**:

```
sudo zypper ref
```

3. Instale o Wine (digite **y** para iniciar a instalação) e execute-o uma vez:

```
sudo zypper in wine
wine64 ipconfig
```

elementary OS

> **Aplica-se a:** elementary OS 5.0 e 5.1

- **elementary OS 5.1:** siga as instruções do Ubuntu.
- **elementary OS 5.0:** não suportado. Em nossos testes, a instalação do Wine deixou o sistema sem conseguir inicializar, por isso não fornecemos instruções para ele.

Syhunt Hybrid 7.2

Solus

> **Aplica-se a:** Solus 4.1

Atenção
O Solus não é suportado. O Wine64 se mostrou instável no Solus 4.1 em nossos testes (erros como `wine client error:6c: write: Bad file descriptor`), tanto com o pacote `wine-stable` quanto com o `wine-devel`.

Se ainda assim quiser tentar:

```
sudo eopkg upgrade openssl
sudo eopkg install wine-devel
wine64 ipconfig
```


Página 52 de 182

Syhunt Hybrid 7.2

Notas de Migração

Leia as notas de todas as versões entre a sua versão atual e a que você está instalando.

Migração para o Syhunt 7.2

- **A interface clássica de desktop foi descontinuada.** A interface de desktop mostrada abaixo não é mais incluída; ela foi substituída pela nova interface web.

- **Mesmos recursos no Linux.** Recursos antes disponíveis apenas no Windows, como a autenticação baseada em Selenium e o rastreador AJAX avançado, agora também estão disponíveis no Linux.
- **O Forensic Scanner está disponível apenas pela CLI.** Use o comando `scanlog`. Uma atualização futura permitirá ver os resultados das varreduras forenses na interface web.
- **Revise seus alvos.** Abra a tela Targets e procure ícones de alerta ao lado de algum alvo. Passe o mouse sobre o ícone para ver quais preferências ou configurações precisam de atenção. Em especial:
  - O método de autenticação **Selenium-Py** está obsoleto. Migre para um dos novos métodos: Smart, Recorder, Selenium-Auto ou Selenium-JS.
  - O **AI-Powered Login** foi descontinuado. Migre para Smart, Recorder ou Selenium-Auto.
- **As extensões Plus agora vêm incluídas.** Não é mais necessário instalar o instalador separado das extensões Plus; o instalador da versão 7.2 já inclui todos os componentes necessários.

Migração para o Syhunt 7.0

No Windows Server, certifique-se de que o **WebView2 Runtime** está instalado. Sem ele, o Syhunt exibe um alerta ao abrir sua interface. Você pode obtê-lo no Evergreen Standalone Installer.

Migração para o Syhunt 6.9.26.1

Esta versão muda o funcionamento das preferências de site no Syhunt Dynamic: diferentes caminhos de URL inicial no mesmo domínio agora podem ter suas próprias preferências. Por exemplo, `https://mydomain/`, `https://mydomain/admin/` e `https://mydomain/members/` passam a ser URLs iniciais distintas, cada uma com suas próprias preferências de site.

A mudança se aplica às preferências definidas ao iniciar uma varredura sob demanda, nas varreduras agendadas, na tela de alvos dinâmicos e pela ferramenta de CLI `scancore`.

Revise suas preferências de site existentes. Se uma URL inicial inclui um caminho (ou seja, não é a raiz do site), suas preferências agora mostrarão os valores padrão, e será necessário recriar configurações anteriores, como credenciais e restrições de rastreamento.

Página 53 de 182

Migração para o Syhunt 6.9.9

- No Linux, baixe e instale esta atualização manualmente, em vez de usar o `scanupdate`.
- Verifique se as varreduras de código agendadas e em pipelines de repositórios Git apontam para o branch correto. Versões anteriores usavam o branch `master` por padrão; a partir da 6.9.9, o nome do branch fica vazio por padrão e o cliente Git escolhe o branch padrão do repositório, a menos que você defina um explicitamente.
- Certifique-se de atender aos novos requisitos de conexão com a Internet.

Migração para o Syhunt 6.9.7 ou posterior

Até a versão 6.9.6, o Syhunt para Windows armazenava os dados do usuário em `%AppData%\Local\Syhunt`. Como parte de melhorias de arquitetura, a versão 6.9.7 e posteriores os armazenam em `%ProgramData%\Syhunt`.

Antes de atualizar, exporte suas preferências para poder importá-las depois:

1. Abra o Syhunt Hybrid 6.9.6 (ou sua versão mais antiga).
2. Escolha ☰ **Menu → Import & Export → Export Preferences**.
3. Salve o arquivo de backup em um local fácil de encontrar, com um nome fácil de lembrar.
4. Instale o Syhunt 6.9.7 ou posterior.
5. Escolha ☰ **Menu → Import & Export → Import Preferences**, selecione o arquivo de backup e clique em **Open**.
6. Todas as configurações da instalação anterior, incluindo as varreduras agendadas, estarão importadas e ativas.

(Opcional) Migração do histórico e dos resultados de sessões

- Para manter suas sessões anteriores, mova os arquivos e pastas de `%AppData%\Local\Syhunt\Sessions` para `%ProgramData%\Syhunt\Sessions`. Elas voltarão a aparecer na tela Past Sessions.
- Se necessário, mova também `%AppData%\Local\Syhunt\Output` para `%ProgramData%\Syhunt\Output`.
- As demais pastas em `%AppData%\Local\Syhunt\` contêm dados temporários e podem ser excluídas com segurança.

Modo AppData forçado

No Windows, o Syhunt 6.9.7.1 e posteriores podem ser forçados a continuar usando a pasta local AppData em vez da ProgramData: crie um arquivo vazio chamado `LocalAppData.json` no local padrão do Syhunt (consulte o guia de Linha de Comando para ver os locais padrão).

> **Atenção**
> Alguns recursos importantes, como o agendador de varreduras, **não** funcionam neste modo.

Syhunt Hybrid 7.2

Ajuste de Desempenho

Como lidar com varreduras DAST longas

A varredura de um site que gera muito conteúdo dinâmico pode levar bastante tempo com o Syhunt Dynamic. Isso não é um bug nem uma limitação, mas consequência de uma decisão de projeto: o Syhunt Dynamic **não** limita a profundidade com que mapeia a estrutura e os pontos de entrada de um site, pois o objetivo é sempre testar o alvo por completo. Alguns sites, porém, são extremamente grandes ou geram uma quantidade praticamente infinita de links, o que se torna um problema quando a varredura precisa terminar dentro de um prazo.

Consideramos que uma varredura DAST de um site com conteúdo muito dinâmico não deve levar mais de **três horas**. Para isso, o Syhunt Dynamic oferece:

- **Varredura incremental**, que reduz o tempo de novas varreduras do mesmo alvo.
- **Um conjunto grande e crescente de regras integradas** que aceleram varreduras desse tipo de site. Novas regras são incluídas nas atualizações do Syhunt toda semana.

Otimização de DAST

Quando as regras integradas não bastam para um site específico, um especialista da Syhunt pode analisá-lo, mediante solicitação, e adicionar regras personalizadas. Chamamos esse processo de *otimização de DAST*; normalmente ele leva apenas algumas horas.

As novas regras aceleram o mapeamento e direcionam o scanner às áreas e pontos de entrada relevantes do site, ignorando conteúdo dinâmico redundante. Nenhum ataque é realizado contra o site durante a otimização, portanto não há risco para sua integridade ou disponibilidade. Ao final do trabalho, uma nova versão do Syhunt com as novas regras é disponibilizada.

Para solicitá-la, abra um ticket de suporte listando as URLs alvo que precisam de otimização de DAST.

Complementando o DAST com SAST

Para sites com conteúdo muito dinâmico, a Syhunt recomenda também fazer a varredura do código-fonte com o Syhunt Code. A análise estática cobre trechos de código difíceis de alcançar pelo rastreamento.

Acelerando varreduras de código

A proteção antivírus em tempo real pode deixar as varreduras de código-fonte bem mais lentas. Consulte Como evitar a interferência do antivírus em pipelines.

Página 55 de 182

Syhunt Hybrid 7.2

Varreduras

Página 56 de 182

Syhunt Hybrid 7.2

Varredura Dinâmica (DAST)

> **Aplica-se a:** Syhunt Dynamic 7.2

Conteúdo

- Como fazer uma varredura dinâmica
- Login manual pelo navegador
- Varredura pela linha de comando
- Varredura de endereços IPv6
- Como impedir que uma vulnerabilidade seja relatada
- Requisitos prévios
- Guias relacionados

Como fazer uma varredura dinâmica

Em uma varredura dinâmica padrão (também chamada de varredura caixa-preta), o Syhunt injeta dados na aplicação web e analisa as respostas para determinar se o código é vulnerável a ataques específicos contra aplicações web.

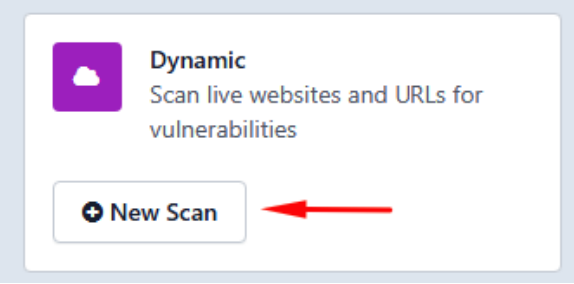
Principais linguagens suportadas

ASP (Classic) · ASP.NET · Java / JSP · JavaScript · Lua · Perl · PHP · Python · Ruby

Executando a varredura

Siga estes passos para fazer uma varredura dinâmica e gerar um relatório de vulnerabilidades:

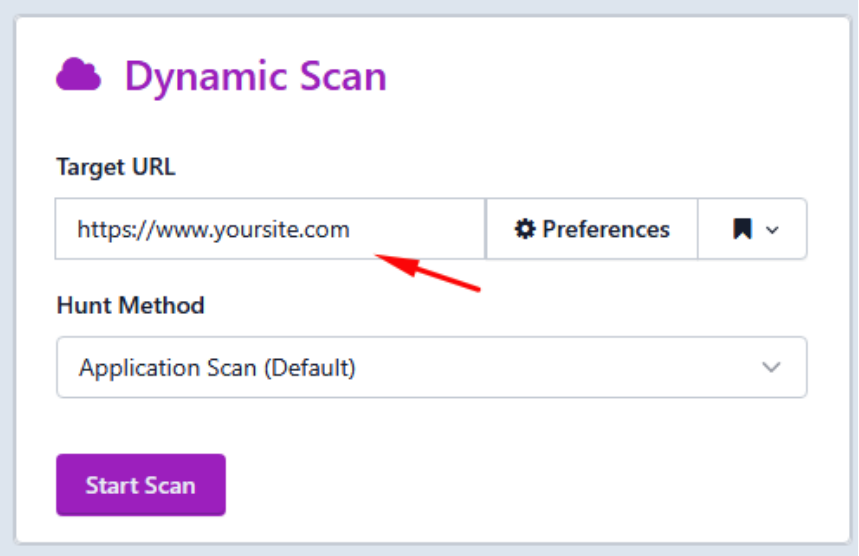
1. ⚠️ Certifique-se de atender aos requisitos prévios e de estar autorizado a fazer a varredura do alvo.
2. Habilite a interface web, caso ainda não tenha feito isso.
3. Na interface web do Syhunt, clique no botão **New Scan** do Dynamic.



Página 57 de 182

Syhunt Hybrid 7.2

4. Informe a URL do site cuja varredura deseja fazer.



5. Selecione um método de varredura. Recomendamos o **Application Scan (Default)**, que procura todas as vulnerabilidades usando as configurações recomendadas. Os métodos são explicados em Métodos de Varredura.
6. *(Opcional)* Clique em **Preferences**. Na aba **Technologies**, selecione as tecnologias usadas pelo site alvo. Na mesma tela, você pode alterar outras preferências específicas do site. Revise as configurações e clique em **Save**.
7. Clique em **Start Scan**.

Ao final da varredura, clique em **View Report** para abrir os resultados como relatório HTML, ou em **Export** para baixá-los no formato de sua preferência.

Login manual pelo navegador

Consulte Varredura com login manual (Sandcat).

Varredura pela linha de comando

Consulte Linha de Comando: varreduras dinâmicas.

Varredura de endereços IPv6

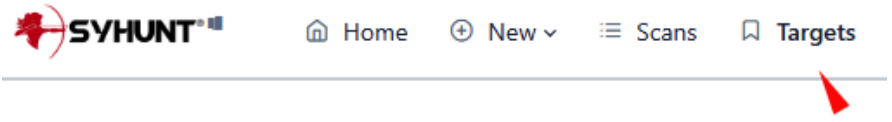
O Syhunt Dynamic suporta totalmente alvos IPv6. Coloque o endereço entre colchetes, por exemplo:

```
http://[2001:4860:0:2001::68]/index.php
```

Como impedir que uma vulnerabilidade seja relatada

Você pode impedir que vulnerabilidades específicas sejam relatadas usando IDs de exclusão (ignore IDs) ou regras de exclusão:

1. Abra **Targets** no menu

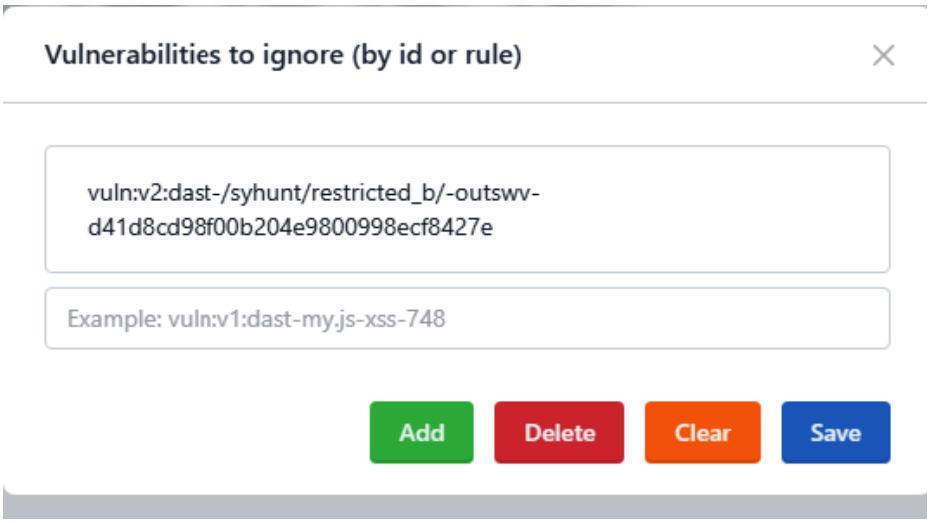


e clique no alvo na lista.

Página 58 de 182

Syhunt Hybrid 7.2

2. Vá para a aba **Manage**.
3. No grupo **Exclusions**, clique em **Vulnerabilities**.
4. Digite o ID ou a regra de exclusão e clique em **Add**.



Cada vulnerabilidade de um relatório termina com seu ID de exclusão. Os IDs de exclusão são a forma mais simples e recomendada de suprimir um resultado. Para cenários mais amplos, você pode escrever regras de exclusão.

Para dúvidas comuns, consulte as Perguntas Frequentes.

Requisitos prévios

1. ⚠️ **Somente administradores de sistema, ou outras pessoas responsáveis pelo alvo, devem usar este software. Não faça varreduras de sites que não estejam sob seu controle direto.**
   - Se precisar fazer a varredura de um site que você não controla, obtenha antes uma autorização por escrito do proprietário ou administrador.
   - Como boa prática, faça backup do código-fonte e do banco de dados do site antes da varredura. Em casos raros, testes de injeção contra um site inseguro e frágil podem poluir o banco de dados ou alterar arquivos, prejudicando o funcionamento do site.
2. Certifique-se de atender aos requisitos de conexão com a Internet.
3. Leia e aceite o EULA do Syhunt antes de iniciar qualquer varredura.

Guias relacionados

- Métodos de Varredura: como escolher o método adequado.
- Varredura Autenticada: varredura de áreas protegidas por login.
- Agendamento de Varreduras: execução automática de varreduras.
- Relatórios: personalização de relatórios.
- Especificações Técnicas: requisitos de sistema.

Página 59 de 182

Syhunt Hybrid 7.2

Varredura Autenticada

> **Aplica-se a:** Syhunt Dynamic 7.2.0.24

Conteúdo

- Introdução
- Métodos de autenticação disponíveis
- Novidades da versão 7.2
- Configurando e testando a autenticação
- Gravando uma sequência de login (Recorder)
- Varredura com login manual (Sandcat)
- Autenticação baseada em Selenium
- Varreduras agendadas com autenticação
- Varreduras autenticadas pela linha de comando
- Personalizando o preenchimento automático de formulários
- Como evitar logout acidental
- Uso de certificados SSL de cliente

Introdução

Desde a versão 7.2, o Syhunt Dynamic suporta uma ampla variedade de métodos de autenticação: **Standard**, **Recorder**, **Selenium-Auto**, **Selenium-JS**, **Smart**, **Basic**, **Bearer**, **Digest**, **NTLM** e **Manual**.

A maioria dos métodos é automatizada: o Syhunt Dynamic faz login usando as credenciais e configurações definidas na tela Target Preferences. Dependendo do método, ele detecta o formulário de login automaticamente, reproduz um login gravado no navegador, executa um script Selenium ou usa um mecanismo de autenticação HTTP. Quando o método permite, o Syhunt Dynamic monitora a sessão autenticada e faz login novamente de forma automática se a sessão for perdida.

A **autenticação manual** inicia uma varredura a partir de uma sessão do navegador em que você já fez login no Edge ou no Chrome, usando a extensão Syhunt Sandcat. O Syhunt Dynamic monitora essa sessão durante toda a varredura e informa se ela foi perdida antes do fim.

Métodos de autenticação disponíveis

Método	Descrição
None	A autenticação fica desativada.
Standard	Login clássico por formulário, usando requisições HTTP diretas, sem sessão de navegador.
Smart	Tenta primeiro o Selenium-Auto e, se necessário, recorre ao Standard.
Recorder	Reproduz uma sequência de login gravada no Edge, no Chrome ou em outro navegador baseado no Chromium.
Pre (Manual)	Usa uma sessão de navegador em que você fez login no Edge ou no Chrome, por meio da extensão Syhunt Sandcat. Diferentemente dos demais métodos, é iniciado pelo Sandcat, e não configurado na tela Target Preferences.
Selenium-Auto	Gera e executa automaticamente um script de login Selenium. Não exige programação.
Selenium-JS	Executa um script de login em Node.js fornecido por você. Por segurança, os scripts só podem ser informados pela CLI.
Bearer	Autentica as requisições com um token bearer. Comum em serviços REST, APIs web, aplicativos mobile e aplicações web modernas.
Basic	Autenticação HTTP Basic.

Página 60 de 182

Método	Descrição
Digest	Autenticação HTTP Digest.
NTLM	Autenticação NTLM, comum em ambientes Windows e Active Directory.

Novidades da versão 7.2

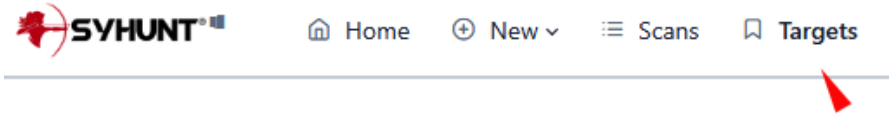
A versão 7.2 introduz o **Syhunt Authenticator 2.0**, uma grande reformulação da estrutura de autenticação do Syhunt Dynamic.

- **Um único módulo para todos os métodos.** Antes, o autenticador cuidava principalmente dos logins baseados em Selenium. Agora ele gerencia todos os métodos de autenticação, oferecendo uma experiência de configuração mais consistente.
- **Um único lugar para configurar.** As configurações que antes ficavam separadas entre autenticação por formulário web e métodos do lado do servidor (Basic, Digest, NTLM) agora estão juntas na tela Target Preferences.
- **Novos métodos:**
  - **Recorder**: grava e reproduz sequências de login no navegador.
  - **Smart**: combina Selenium-Auto e Standard para maior compatibilidade.
  - **Selenium-Auto**: gera e executa scripts de login Selenium sem que você precise programar.
  - **Selenium-JS**: executa scripts de login personalizados escritos em Node.js.
- **Métodos removidos.** Os métodos antigos **Selenium-Python** e **AI-Powered** foram removidos. Você não precisa mais instalar e manter um ambiente Python separado e dependências de terceiros, o que também reduz a exposição a componentes externos e a ataques à cadeia de suprimentos de software. O Selenium-Auto substitui o login AI-Powered e funciona de forma muito parecida, sem exigir Python nem integração com IA.

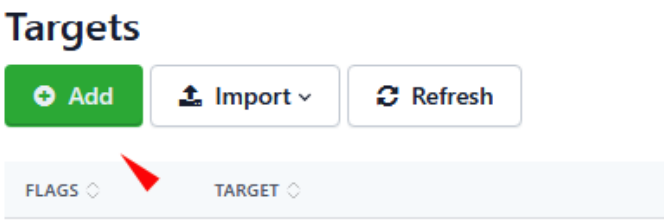
O resultado é uma configuração de autenticação mais simples, mais segura e mais bem integrada ao fluxo de varredura do Syhunt Dynamic.

Configurando e testando a autenticação

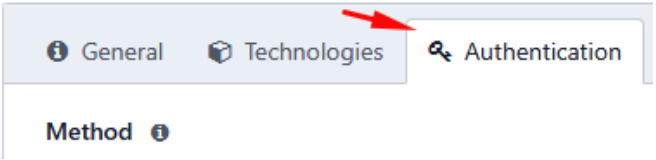
1. Abra a tela **Targets** na interface web.



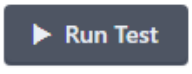
2. Adicione a URL à lista de alvos, caso ainda não esteja nela.



3. Abra o alvo e vá para a aba **Authentication**.



4. Selecione o método de autenticação.
5. Informe as credenciais.
6. Clique em **Run Test** para verificar se a autenticação funciona.



Syhunt Hybrid 7.2

- Em caso de sucesso, aparece a mensagem **Success** e uma captura da página após o login.

- Em caso de falha, aparece a mensagem **Failed** e um log de depuração.

Para instruções específicas de cada método, consulte Recorder, Selenium ou Login manual.

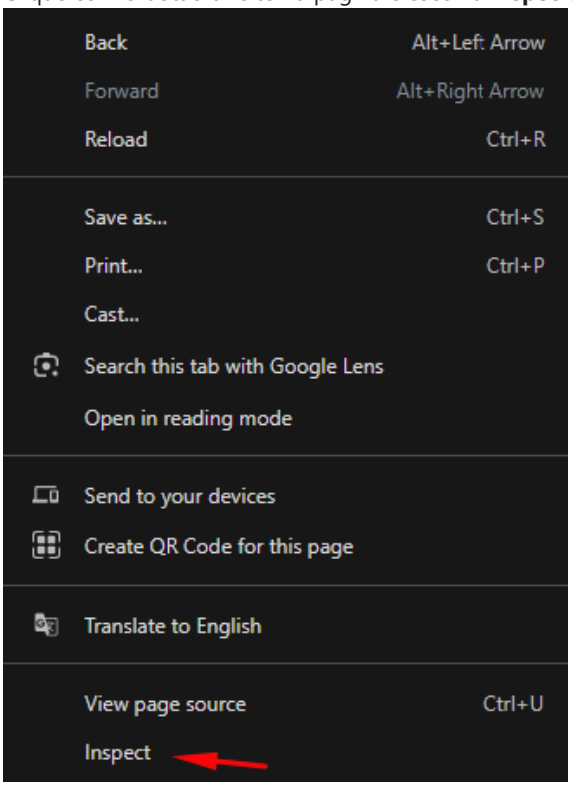
Gravando uma sequência de login (Recorder)

Com o método **Recorder**, o Syhunt Dynamic faz login reproduzindo um login que você gravou no Edge, no Chrome ou em outro navegador baseado no Chromium. Em vez de escrever um script, você grava a si mesmo fazendo login, exporta a gravação como JSON e cola o conteúdo na tela Target Preferences. O Syhunt Dynamic reproduz os passos gravados sempre que precisar fazer login durante uma varredura.

> **Observação**
> **Usuários de Linux:** não use a tecla Backspace para corrigir erros de digitação durante a gravação. Se digitar algo errado, recomece a gravação. Em nossos testes, gravações com pressionamentos de Backspace às vezes falharam ao ser reproduzidas no Linux.

1. Grave o login

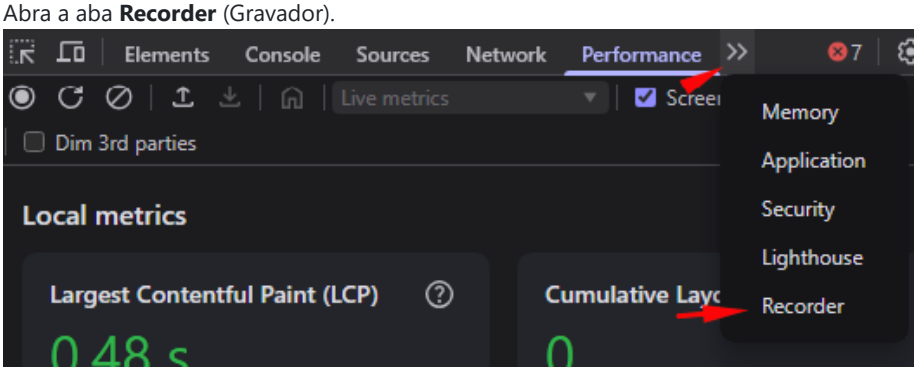
1. No Edge, no Chrome ou em outro navegador baseado no Chromium, acesse o site cuja varredura deseja fazer.
2. Clique com o botão direito na página e escolha **Inspecionar** (no final do menu).



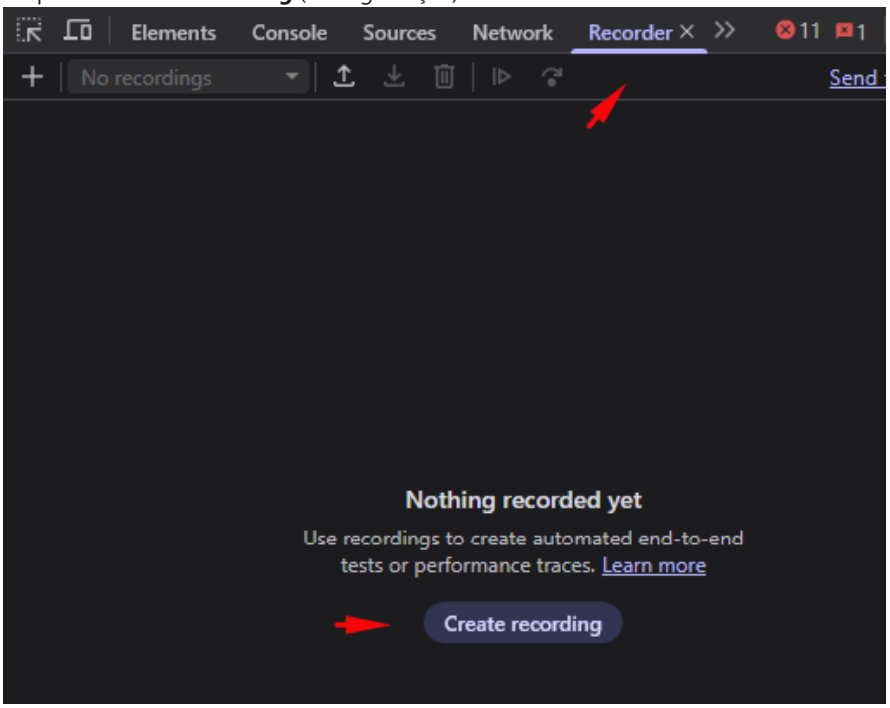
Página 62 de 182

Syhunt Hybrid 7.2

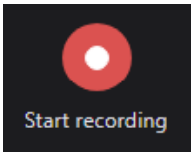
3. Abra a aba **Recorder** (Gravador).



4. Clique em **Create recording** (Criar gravação).

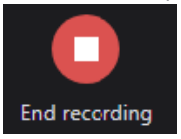


5. Informe um nome para a gravação e clique no botão vermelho **Start recording** (Iniciar gravação).

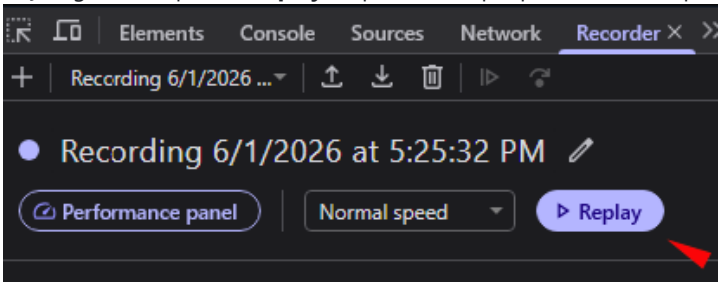


6. Faça login normalmente.

7. Ao terminar, clique em **End recording** (Encerrar gravação).



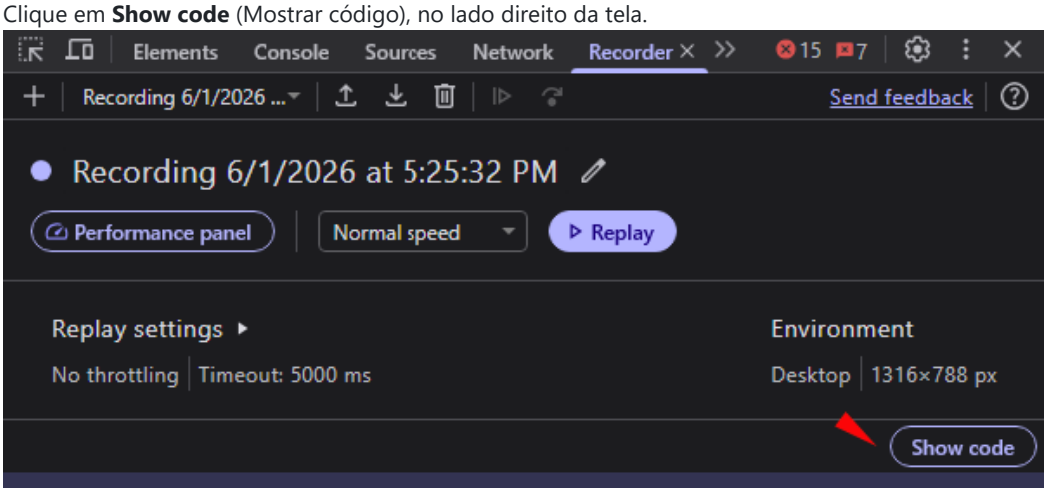
8. Faça logout e clique em **Replay** (Repetir), no topo, para confirmar que a gravação é reproduzida corretamente.



Página 63 de 182

Syhunt Hybrid 7.2

9. Clique em **Show code** (Mostrar código), no lado direito da tela.

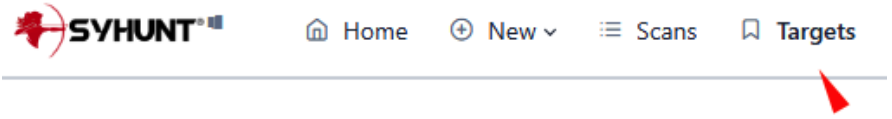


10. Selecione o código JSON manualmente, clique com o botão direito sobre ele e escolha **Copiar**.

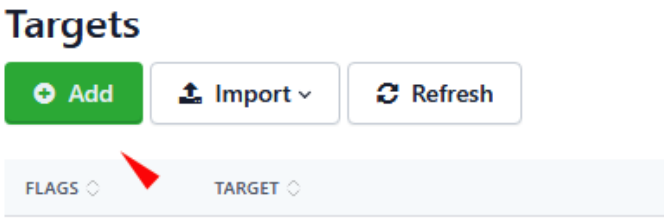
> **Dica**
> Não use *Selecionar tudo* seguido de *Copiar*: um pequeno bug da interface do Chrome impede que o JSON inteiro seja selecionado.

2. Adicione a gravação ao Syhunt

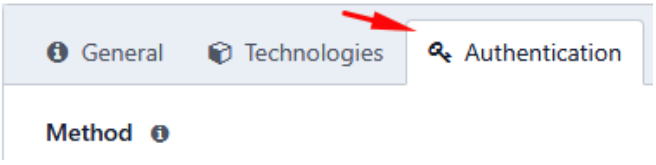
1. Abra a tela **Targets** na interface web.



2. Adicione a URL de login à lista de alvos, caso ainda não esteja nela.



3. Abra o alvo e vá para a aba **Authentication**.

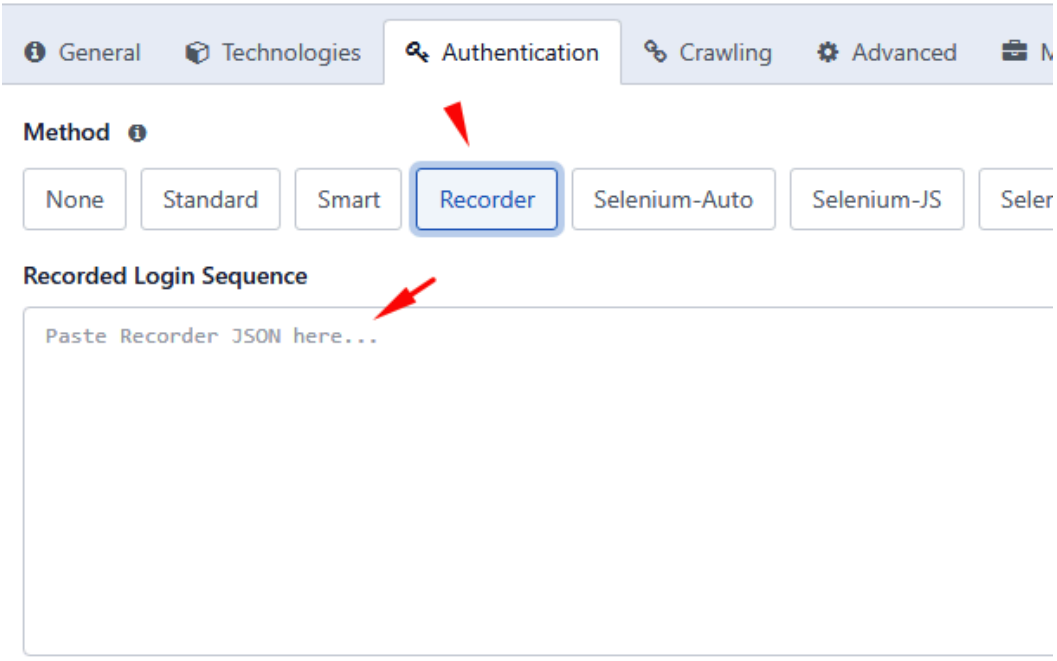


4. Selecione **Recorder**.

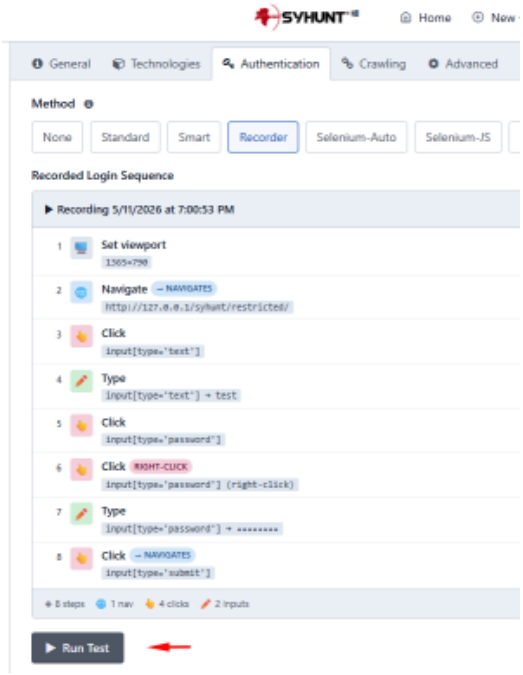
Página 64 de 182

Syhunt Hybrid 7.2

5. Cole o JSON no campo **Recorded Login Sequence**.



6. Clique em **Run Test** para confirmar que o Syhunt consegue reproduzir a gravação.



Varredura com login manual (Sandcat)

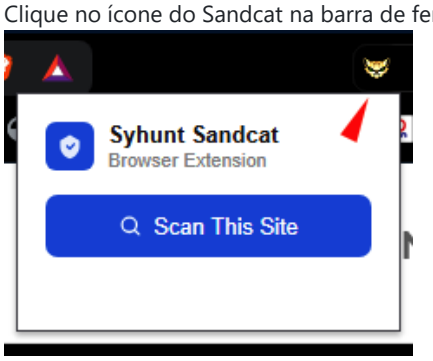
Com a extensão Syhunt Sandcat instalada no navegador (Edge, Chrome, Brave ou outro navegador baseado no Chromium), você pode iniciar uma varredura diretamente pelo navegador, usando a sessão estabelecida ao fazer login no site alvo.

1. Certifique-se de que a extensão Syhunt Sandcat está instalada e habilitada.
2. No navegador, acesse o site cuja varredura deseja fazer.
3. Faça login com suas credenciais.

Página 65 de 182

Syhunt Hybrid 7.2

4. Clique no ícone do Sandcat na barra de ferramentas do navegador e depois em **Scan This Site**.



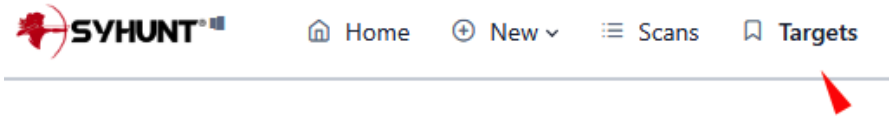
5. Uma nova tela é aberta. A mensagem verde **Session Ready** confirma que a sessão foi capturada.
6. Clique em **Start Scan**.

A seção de detalhes da sessão no relatório informa se a sessão iniciada manualmente se manteve do início ao fim da varredura (**Authenticated Session Maintained: Yes**).

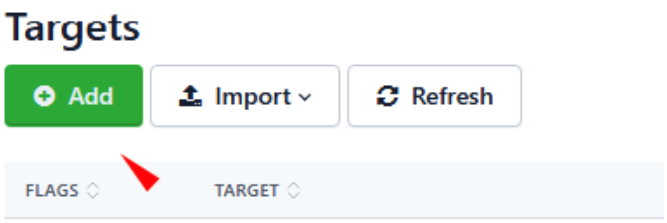
Autenticação baseada em Selenium

1. Configure o alvo

1. Abra **Targets** no menu superior.



2. Adicione a URL à lista de alvos, caso ainda não esteja nela.



3. Clique na URL. A tela Target Preferences é aberta.
4. Na aba **Authentication**, selecione **Selenium-Auto** ou **Selenium-JS**. O Selenium-Auto gera o script para você; o Selenium-JS executa um script escrito por você.
5. Informe o usuário e a senha.
6. Clique em **Save**. A janela Target Preferences é fechada.
7. Se escolheu **Selenium-JS**, informe seu script pela CLI, por exemplo:

```
scancore -tg:http://yourtargetapp.com -prefset:dynamic.formauth.script.seleniumjs.encrypted -
fromfile:caminho/para/myseleniumscript.js
```

Ao escrever o script:

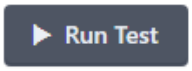
- Comece pela parte que preenche e envia o formulário de login.
- Não chame `driver.get()` no início nem `driver.quit()` no final; o Syhunt cuida de ambos.
- Use os marcadores `@user@` e `@pass@` para o usuário e a senha informados na aba Authentication, por exemplo `username_field.sendKeys('@user@')` e `password_field.sendKeys('@pass@')`.

2. Teste

Página 66 de 182

Syhunt Hybrid 7.2

1. Clique novamente na URL configurada e depois em **Run Test**.



2. Se aparecer **Successful login** e uma captura da página após o login no topo da tela, a configuração está pronta para as próximas varreduras.
3. Se continuar falhando após várias tentativas, ajuste seu script Selenium e tente novamente.

Os relatórios de varreduras autenticadas incluem, no topo, uma captura do alvo após o login e a indicação de se a sessão se manteve do início ao fim.

Varreduras agendadas com autenticação

1. Abra a tela **Scans** e clique em **Scheduled Scans**.
2. Clique em uma varredura dinâmica agendada e depois no botão **Preferences**, ao lado do campo do alvo.
3. Na aba **Authentication**, escolha o método de autenticação e informe as credenciais.

Varreduras autenticadas pela linha de comando

Há duas abordagens:

- **A.** Passar as credenciais ao `scanurl` ao iniciar a varredura.
- **B.** Armazenar as credenciais antecipadamente com o `scancore`.

O método B é mais seguro, pois a senha não fica exposta na linha de comando do `scanurl`.

```
-- Método A: autenticação por formulário informada ao iniciar a varredura.
-- As credenciais não ficam armazenadas nas configurações do Dynamic.
scanurl http://127.0.0.1 -atype:Form -auser:yourusername -apass:yourpassword

-- Método B: autenticação armazenada para um site específico.
-- As credenciais ficam salvas nas configurações do Dynamic e são reutilizadas em todas as varreduras deste alvo.
-- Tipos disponíveis: None, Smart, Standard, Basic, Bearer, Digest e Selenium-Auto.
-- None desativa a autenticação.
scancore -tg:http://127.0.0.1 -prefset:dynamic.formauth.type -v:Standard
scancore -tg:http://127.0.0.1 -prefset:dynamic.formauth.username -v:myuser
scancore -tg:http://127.0.0.1 -prefset:dynamic.formauth.password -vsecret
-- Para autenticação Bearer, defina o token:
scancore -tg:http://127.0.0.1 -prefset:dynamic.formauth.token -vsecret
scanurl http://127.0.0.1
```

O `-vsecret` pede o valor (duas vezes, com a digitação mascarada), para que ele nunca apareça na linha de comando. Consulte Preferências.

Personalizando o preenchimento automático de formulários

> **Observação**
> Isto se aplica aos métodos **Standard**, **Smart** e **Selenium-Auto** e destina-se principalmente a cenários avançados.

O Syhunt Dynamic detecta e preenche automaticamente os campos de formulários de login na maioria das aplicações. Só é preciso configurar os campos manualmente quando o formulário usa nomes de campo fora do padrão, controles personalizados ou campos extras que não podem ser identificados automaticamente.

Por exemplo, este formulário tem um campo oculto extra e nomes fora do padrão:

```
<input name="ClientUTBox" id="ClientUTBox" type="hidden" value="1234">
<input name="ClientUNBox" id="ClientUNBox" type="text" class="InputBox"/>User Name
```

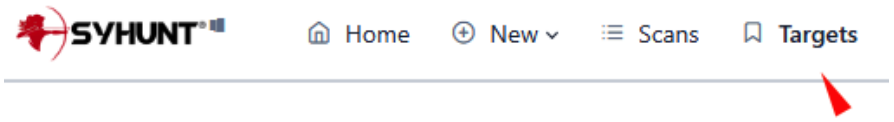

Página 67 de 182

Syhunt Hybrid 7.2

```
<input name="ClientPWBox" id="ClientPWBox" type="password" class="InputBox">Password
```

Para indicar ao Syhunt como preenchê-lo:

1. Abra **Targets** no menu superior.



2. Clique na URL do alvo (ela já deve estar na lista). A tela Target Preferences é aberta.
3. Na aba **Authentication**, clique em **Optional** para exibir as opções adicionais.
4. Clique em **Custom Fields** e adicione estas linhas:

```
ClientUTBox=1234
ClientUNBox=@syhunt_web_form_username
ClientPWBox=@syhunt_web_form_password
```

5. Clique em **Save** para salvar os campos personalizados e depois em **Save**, na parte inferior da tela.

Como alternativa, defina os mesmos campos pela linha de comando:

```
scancore -tg:"[targeturl]" -prefset:dynamic.formauth.fields.encrypted -v:[ClientUTBox=1234],
[ClientUNBox=@syhunt_web_form_username],[ClientPWBox=@syhunt_web_form_password]
```

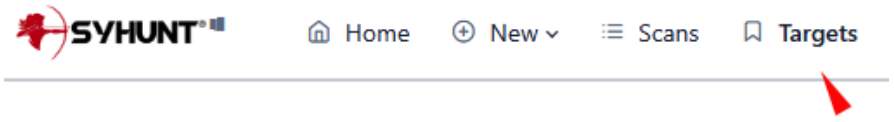
Os valores que começam com `@` são variáveis internas: elas inserem nesses campos o usuário e a senha informados na tela Target Preferences. Ao terminar, clique em **Run Test** para confirmar que o login funciona com as credenciais e os campos personalizados configurados.

Como evitar logout acidental

O Syhunt Dynamic detecta a maioria das páginas de logout e as evita durante o rastreamento. Se acabar acessando uma delas, normalmente ele consegue fazer login de novo automaticamente e continuar.

Para aplicações com URLs de logout personalizadas ou mecanismos de logout incomuns, você pode listar as páginas de logout para que o rastreador as evite completamente:

1. Abra a tela **Targets**.



2. Abra o alvo.
3. Vá para a aba **Crawling**.
4. Em **Exclusions**, clique em **Logout URLs** e adicione sua URL de logout, por exemplo `/getmeout.php`.
5. Clique em **Save**. A janela é fechada.

Uso de certificados SSL de cliente

Se o alvo exigir certificados SSL de cliente, configure-os da seguinte forma:

1. Abra **Admin** → **Connectors**.
2. Clique em **Add**.
3. Defina **Type** como **Web Certificate**, informe um nome para o certificado e clique em **Create**.
4. Clique no novo certificado na lista. A tela Web Certificate é aberta.

Página 68 de 182

Syhunt Hybrid 7.2

5. Selecione os arquivos do certificado e clique em **Save**.
6. Abra a tela **Targets** e clique no site que precisa do certificado. A tela Target Preferences é aberta.
7. Na aba **Advanced**, em **Certificate**, selecione o certificado na lista.

Página 69 de 182

Syhunt Hybrid 7.2

Varredura de Código-Fonte (SAST/MAST)

> **Aplica-se a:** Syhunt Code 7.2

Conteúdo

- Como fazer uma varredura de código
- Linguagens suportadas
- Varredura pela linha de comando
- Como evitar a interferência do antivírus em pipelines
- Como impedir que uma vulnerabilidade seja relatada
- Guias relacionados

Como fazer uma varredura de código

A varredura caixa-branca do Syhunt (varredura de código-fonte) revela muitas classes de vulnerabilidades e destaca as áreas do código que precisam de revisão. Sua análise estática detecta uma quantidade muito grande de tipos de vulnerabilidades, incluindo o CWE Top 25 de 2025 (as fraquezas de software mais perigosas) e os riscos do OWASP Mobile Top 10, em diversas linguagens de programação web e mobile.

Linguagens suportadas: web

Linguagem	Frameworks e variantes
C#	Blazor, Razor Pages, ASP.NET Core MVC, ASP.NET Web Forms
ASP Classic	VBScript e JavaScript
Dart	
Java	JEE / JSP
JavaScript	Lado do cliente e do servidor, Node.js, Angular, AngularJS, React, Express.js, Koa.js
Kotlin	Ktor
Lua	ngx_lua, mod_lua, CGILua, Lua Pages
Object Pascal	Delphi XE e anteriores, Lazarus, DWS
Perl	Mojolicious, PSGI/Plack, CGI
PHP	
Python	CGI, Django, mod_python, WSGI
Ruby	Rails, ERB
TypeScript	Lado do cliente e do servidor, Node.js, Angular
VB	VB.NET, ASP.NET Web Forms

Linguagens suportadas: mobile

Linguagem	Plataforma
Java	Android
Kotlin	Android
Swift	iOS

Página 70 de 182

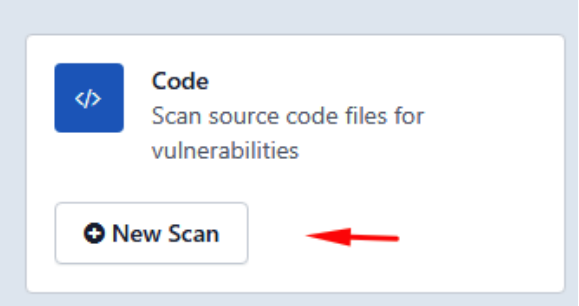
Syhunt Hybrid 7.2

Linguagem	Plataforma
Objective-C, C e C++	iOS
C#	.NET MAUI, Xamarin (Android/iOS)
Dart	Flutter (Android/iOS)
Object Pascal	Delphi XE
JavaScript	Incluindo Node.js, Angular, AngularJS, Express.js, Koa.js

Executando a varredura

Siga estes passos para fazer uma varredura de código-fonte e gerar um relatório de vulnerabilidades:

1. Habilite a interface web, caso ainda não tenha feito isso.
2. Na interface web do Syhunt, clique no botão **New Scan** do Code.



3. Informe a URL do projeto ou o arquivo a ser verificado.
4. *(Opcional)* Se informou a URL de um projeto, também pode informar o nome de um branch. Se deixar em branco, será usado o branch principal.
5. Selecione um método de varredura. Recomendamos o **Application Code Scan (Default)**, que procura todas as vulnerabilidades usando as configurações recomendadas. Os métodos são explicados em Métodos de Varredura.
6. Clique em **Start Scan**.

Ao final da varredura, clique em **View Report** para abrir os resultados como relatório HTML, ou em **Export** para baixá-los no formato de sua preferência.

Varredura pela linha de comando

Consulte Linha de Comando: varreduras de código.

Como evitar a interferência do antivírus em pipelines

Ao analisar um código grande, o Syhunt abre todos os arquivos do repositório. Se a proteção antivírus em tempo real estiver ativada, o antivírus também verifica cada um desses arquivos, o que pode deixar a varredura consideravelmente mais lenta.

Em nossos testes em uma máquina com o Microsoft Defender, a varredura de um repositório PHP com cerca de 10.000 arquivos levou 10 minutos a mais e usou 25% mais CPU por causa do Antimalware Service (`MsMpEng.exe`). Em um pipeline de CI/CD, esses 10 minutos fazem diferença.

A solução é adicionar uma exclusão do tipo **Processo** no Defender. Uma exclusão de processo impede que o antivírus verifique os arquivos abertos pelo processo do Syhunt; já uma exclusão de *arquivo* exclui apenas o próprio executável.

Para adicionar uma exclusão de processo no Microsoft Defender:

Página 71 de 182

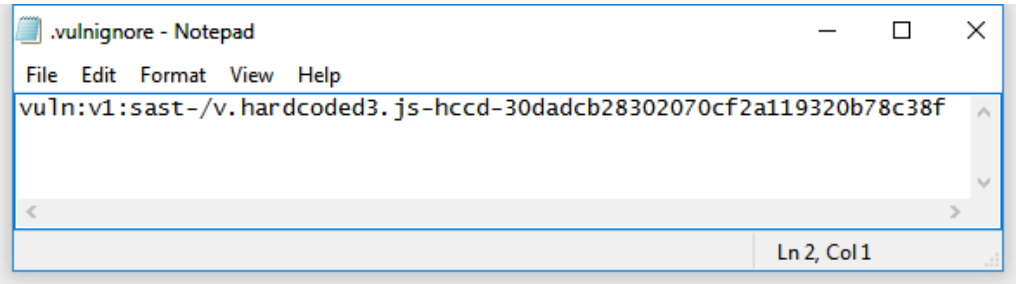
Syhunt Hybrid 7.2

1. Abra **Iniciar** → **Configurações**.
2. Em **Privacidade e segurança**, selecione **Proteção contra vírus e ameaças**.
3. Em **Configurações de proteção contra vírus e ameaças**, selecione **Gerenciar configurações**.
4. Em **Exclusões**, selecione **Adicionar ou remover exclusões** e escolha **Processo**.
5. Informe o caminho completo do scanner do Syhunt Code, `C:\Program Files\Syhunt Hybrid\ScanCode.exe`, e clique em **Adicionar**.

Se você usa outro antivírus, consulte a documentação dele para encontrar a configuração equivalente.

Como impedir que uma vulnerabilidade seja relatada

Para suprimir resultados específicos, crie um arquivo `.vulnignore` na raiz do repositório ou da pasta verificada. Coloque um ID ou regra de exclusão por linha.



Cada vulnerabilidade de um relatório termina com seu ID de exclusão. Os IDs de exclusão são a forma mais simples e recomendada de suprimir um resultado. Para cenários mais amplos, você pode escrever regras de exclusão.

Guias relacionados

- Métodos de Varredura: como escolher o método adequado.
- Agendamento de Varreduras: execução automática de varreduras.
- Relatórios: personalização de relatórios.
- Especificações Técnicas: requisitos de sistema.

Página 72 de 182

Syhunt Hybrid 7.2

Varredura de APIs Web

> **Aplica-se a:** Syhunt API 7.2

Conteúdo

- Como funciona a varredura de APIs
- Formatos de especificação suportados
- Executando uma varredura de API
- Especificações de demonstração
- Como impedir que uma vulnerabilidade seja relatada
- Métodos de varredura para APIs
- Agendamento de varreduras de APIs
- Requisitos prévios
- Hospedagem e envio de especificações
- Proteções contra varreduras acidentais
- Requisitos de formato das especificações

Como funciona a varredura de APIs

Uma varredura de API funciona de forma muito parecida com uma varredura dinâmica tradicional: o Syhunt injeta dados de teste nos endpoints da API e analisa as respostas para descobrir se a API é vulnerável a ataques específicos no nível da aplicação.

A diferença é o ponto de partida. Uma varredura dinâmica começa a partir de uma URL e rastreia o site; uma varredura de API começa a partir de uma **especificação da API** (uma URL ou um arquivo). Em vez de rastrear, o Syhunt mapeia a API a partir da especificação e então executa uma etapa de **validação de esquema**, uma forma de teste funcional que confirma que a API existe e se comporta conforme especificado antes do início dos testes de segurança. Se encontrar uma divergência importante, a varredura é interrompida automaticamente e um alerta é exibido, para que você não receba resultados enganosos.

Formatos de especificação suportados

- API Blueprint
- OpenAPI 2 e 3
- Swagger 1, 2 e 3
- Postman Collection 1, 2 e 2.1
- GraphQL (com anotações REST)
- I/O Docs
- Google Discovery
- RAML
- WADL

Antes da varredura, o Syhunt verifica se a URL da especificação é válida e está em um formato suportado, e rejeita tudo o que for incompatível ou inválido. A licença básica suporta os principais formatos (OpenAPI 2 e 3, Swagger 1 a 3 e API Blueprint), que juntos representam mais de 95% das especificações de API em uso atualmente. Outras licenças podem incluir GraphQL, I/O Docs, Google Discovery, RAML e WADL.

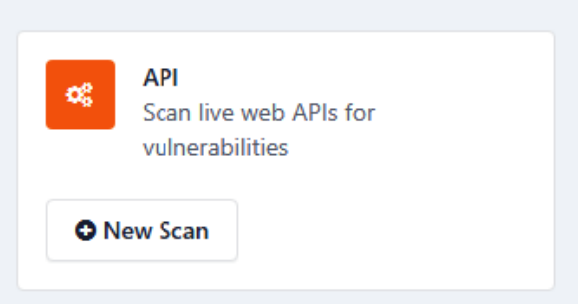
Como as APIs são caixas-pretas, o Syhunt precisa de uma especificação para fazer a varredura: ela fornece a estrutura e as definições de endpoints que orientam os testes. O Syhunt tem ampla compatibilidade, mas recomendamos o API Blueprint. Sua sintaxe Markdown limpa e legível, seus bons recursos de documentação e a facilidade de integração aos fluxos de desenvolvimento e teste combinam com o foco do Syhunt em facilidade de uso.

Página 73 de 182

Syhunt Hybrid 7.2

Executando uma varredura de API

1. ⚠️ Certifique-se de atender aos requisitos prévios e de estar autorizado a fazer a varredura da API alvo. O Syhunt inclui proteções contra a varredura de APIs fora do seu controle direto.
2. Habilite a interface web, caso ainda não tenha feito isso.
3. Na interface web, clique no botão **New Scan** da API.



4. Informe a URL da especificação da API, por exemplo `http://mysite/api/openapi.json`. Para experimentar, use uma das especificações de demonstração.

> **Observação**
> Você receberá um erro **Invalid target** se informar a URL da própria API em vez da URL da especificação, ou se a especificação for inválida.

5. Selecione um método de varredura. Recomendamos o **Application Scan (Default)**, que procura todas as vulnerabilidades usando as configurações recomendadas. Consulte Métodos de Varredura.
6. *(Opcional)* Clique em **Preferences**, ao lado da URL alvo, para personalizar as configurações da API.
7. Clique em **Start Scan**.

Você pode clicar em **View Report** a qualquer momento durante a varredura para ver os resultados obtidos até então em um relatório HTML.

Para fazer a varredura pela linha de comando, consulte Linha de Comando: varreduras de API.

Especificações de demonstração

A interface web inclui APIs vulneráveis simuladas para demonstração e testes. Aponte o scanner para uma das URLs de especificação abaixo. Essas especificações referenciam apenas endpoints acessíveis localmente e não contêm vulnerabilidades reais; elas apenas simulam um comportamento vulnerável, para que você possa testar e aprender com segurança.

Formato	URL da especificação
API Blueprint	`http://spec.syhunt.test/vulnapisim.apib`
OpenAPI 3	`http://spec.syhunt.test/vulnapisim_openapi3.json`
Swagger 2	`http://spec.syhunt.test/vulnapisim_swagger2.json`
Postman 2.1	`http://spec.syhunt.test/vulnapisim_postman2.1.json`
Postman 1	`http://spec.syhunt.test/vulnapisim_postman1.json`

Página 74 de 182

Syhunt Hybrid 7.2

Como impedir que uma vulnerabilidade seja relatada

Resultados de varreduras de API podem ser suprimidos da mesma forma que nas varreduras dinâmicas, usando os IDs de exclusão exibidos nos relatórios.

Métodos de varredura para APIs

A maioria dos métodos de varredura disponíveis para varreduras dinâmicas também funciona em varreduras de API, com uma diferença importante no método **Top 10**:

- Para aplicações web, ele cobre o **OWASP Top 10 para Aplicações Web**: injeção, quebra de autenticação, cross-site scripting e assim por diante.
- Para APIs, ele cobre **tanto** o OWASP Top 10 para Aplicações Web **quanto** o **OWASP API Security Top 10**, que acrescenta riscos específicos de APIs, como quebra de autorização no nível de objeto (BOLA), exposição excessiva de dados, ausência de limitação de taxa e configurações incorretas específicas de APIs.

Isso reflete um princípio da Syhunt: o OWASP Top 10 clássico continua muito relevante para APIs, pois muitos riscos de aplicações web também aparecem no nível da API. Aplicar as duas listas resulta em uma avaliação Top 10 realmente abrangente, que cobre tanto os riscos que as APIs herdam das aplicações web quanto os exclusivos das APIs.

Métodos que não se aplicam a APIs ficam indisponíveis. Se você iniciar uma varredura de API com um deles, o Syhunt a interrompe e exibe um alerta.

Agendamento de varreduras de APIs

Varreduras de API são agendadas da mesma forma que as varreduras dinâmicas. Consulte Agendamento de Varreduras.

Requisitos prévios

1. ⚠️ **É obrigatória uma especificação da API (URL ou arquivo).** Como as APIs são caixas-pretas, uma varredura precisa e completa é impossível sem ela: a especificação define os endpoints, parâmetros e comportamentos esperados que orientam os testes.
    - Nem sempre as equipes de segurança têm acesso às especificações das APIs. Nesse caso, peça-as à equipe de desenvolvimento antes de começar. Trate qualquer API sem documentação adequada ou especificação como um risco de segurança, pois ela não pode ser testada, revisada ou monitorada de forma eficaz.
2. ⚠️ **Somente administradores de sistema, ou outras pessoas responsáveis pelo alvo, devem usar este software. Não faça varreduras de APIs que não estejam sob seu controle direto.**
    - Se precisar fazer a varredura de uma API que você não controla, obtenha antes uma autorização por escrito do proprietário ou administrador.
    - O Syhunt inclui proteções para evitar a varredura acidental de APIs fora do controle da sua organização.
    - Como boa prática, faça backup do código-fonte e do banco de dados da API antes da varredura. Em casos raros, testes de injeção contra uma API insegura e frágil podem poluir o banco de dados ou alterar arquivos, prejudicando o funcionamento da API.
3. Certifique-se de atender aos requisitos de conexão com a Internet.
4. Leia e aceite o EULA do Syhunt antes de iniciar qualquer varredura.

Hospedagem e envio de especificações

Desde o Syhunt 7.1.11, você pode informar o link de uma especificação hospedada em qualquer lugar ou enviar um arquivo de especificação.

Página 75 de 182

Origem	Como funciona	Inclusões (arquivos referenciados)
Hospedada	Informe a URL da especificação.	Suportadas, desde que todos os arquivos incluídos estejam no mesmo caminho da especificação principal, ou abaixo dele.
Enviada	O Syhunt disponibiliza o arquivo em uma URL interna, como `http://spec.syhunt.test/<arquivo.ext>`.	Não suportadas. O arquivo precisa ser autossuficiente.
Local (somente Windows)	Coloque o arquivo em `C:\ProgramData\Syhunt\UserData\Specs`. Ele fica disponível em `http://localspec.syhunt.test/<arquivo.ext>`, acessível apenas pela rede interna, para reduzir a exposição.	Suportadas. O Syhunt busca os arquivos incluídos automaticamente.

Proteções contra varreduras acidentais

O Syhunt interrompe automaticamente uma varredura de API, a menos que uma destas condições seja atendida:

1. O host da API referenciado na especificação está em uma rede interna (uma faixa de IP privada ou um endereço de loopback).
2. O domínio da API foi explicitamente liberado ao ser adicionado à lista de alvos. Por exemplo, se a especificação aponta para `api.mydomain.com`, adicione `mydomain.com` à lista de alvos:
    - na interface web, em `http://127.0.0.1:8017/syhunt/?n=targets`; ou
    - pela CLI: `scancore -tg:https://mydomain.com -prefset:enabled -v:true`

Se nenhuma das condições for atendida, uma mensagem de erro explica o problema.

Essa proteção impede que usuários façam, sem querer, a varredura de APIs externas ou de terceiros que sua organização não controla. Ela mantém as varreduras dentro do escopo autorizado, protegendo tanto o usuário quanto sistemas externos de testes não intencionais ou não autorizados.

Requisitos de formato das especificações

O Syhunt se esforça bastante para aceitar especificações fora do padrão ou escritas de forma pouco rigorosa, mas especificações que se afastam demais do formato oficial podem ser rejeitadas, e a Syhunt não se compromete a suportar especificações malformadas ou fora da conformidade. Se a sua for rejeitada, corrija-a ou use uma ferramenta de terceiros para colocá-la em conformidade, ou converta-a para API Blueprint para obter a melhor compatibilidade e estabilidade.

Requisitos de sistema

Consulte as Especificações Técnicas.

Syhunt Hybrid 7.2

Métodos de Varredura

> **Aplica-se a:** Syhunt Hybrid 7.0.11

Um *método de varredura* (hunt method) define o que o Syhunt procura e com que intensidade ele testa. O **Application Scan** é o padrão; escolha outro método quando precisar de uma varredura mais focada ou mais profunda.

Comparação

Método de varredura	Nome na CLI	Tipo	Força bruta	Injeção	DoS	Demorado
Application Scan (padrão)	`appscan`	▣	S	S	S	N
Application Scan (foco no lado do servidor)	`appscanss`	▣	S	S	S	N
Structure Brute Force	`structbf`	■	S (profunda)	N	N	S (muito)
Old & Backup Files	`fileold`	■	S	N	N	S
Fault Injection	`faultinj`	▣	N	S	S	N
OWASP Top 10	`top10`	▣	N	P (Top 10)	S	N
OWASP Mobile Top 10	`top10mob`	□	N	P (Mobile Top 10)	N	N
CWE Top 25	`top25cwe`	▣	N	P (Top 25)	S	N
OWASP PHP Top 5	`top5php`	▣	N	P (Top 5)	N	N
Cross-Site Scripting	`xss`	▣	N	P (XSS)	N	N
SQL Injection	`sqlinj`	▣	N	P (SQL)	N	N
File Inclusion	`fileinc`	▣	N	P (FI)	N	N
Unvalidated Redirects	`unvredir`	▣	N	P (UR)	N	N
Malware Content	`malscan`	▣	P (malware)	P (malware)	N	N
Passive	`passive`	▣	N	N	N	N
Spider Only	`spider`	■	N	N	N	N
Complete Scan	`complete`	▣	S	S	S	S (muito)
Complete Scan, No DoS	`compnodos`	▣	S	S	N	S (muito)
Complete Scan, Paranoid	`comppnoid`	▣	S (profunda)	S	S	S (muito)

S = sim, **N** = não, **P** = parcial (limitado à categoria indicada).

Tipo de teste

- ▣ Híbrido (caixa-cinza): dinâmico e de código
- ■ Somente dinâmico (caixa-preta)
- □ Somente de código (caixa-branca)

Demorado: *S* significa que o método executa verificações e mutações de ataque adicionais, e que o número de verificações cresce com a quantidade de diretórios encontrados durante o rastreamento.

Descrição dos métodos

Página 77 de 182

Syhunt Hybrid 7.2

Application Scan

Encontra falhas em aplicações web próprias, em software de servidor web e em componentes de terceiros. Rastreia o site e ataca tanto a estrutura do site quanto as aplicações, incluindo testes de injeção de falhas como XSS, injeção de SQL e inclusão de arquivos.

Structure Brute Force

Procura por:

- Scripts vulneráveis comuns
- Arquivos comuns
- Arquivos personalizados (definidos pelo usuário)
- Exposição de bancos de dados
- Backdoors baseados na web

O número de verificações cresce com a quantidade de diretórios encontrados durante o rastreamento.

Old & Backup Files

Testa extensões de backup e de arquivos antigos em cada item da estrutura mapeada do site.

OWASP Top 10

Procura especificamente os riscos do OWASP Top 10 (2017):

1. A1: Injection (Injeção)
2. A2: Broken Authentication (Quebra de autenticação)
3. A3: Sensitive Data Exposure (Exposição de dados sensíveis)
4. A4: XML External Entities (XXE)
5. A5: Broken Access Control (Quebra de controle de acesso)
6. A6: Security Misconfiguration (Configuração de segurança incorreta)
7. A7: Cross-Site Scripting (XSS)
8. A8: Insecure Deserialization (Desserialização insegura)
9. A9: Using Components with Known Vulnerabilities (Uso de componentes com vulnerabilidades conhecidas)
10. A10: Insufficient Logging & Monitoring (Registro e monitoramento insuficientes)

CWE Top 25

Procura especificamente as CWE Top 25 Most Dangerous Software Errors de 2019.

OWASP PHP Top 5

Procura especificamente as cinco principais vulnerabilidades de PHP segundo a OWASP:

1. Execução remota de comandos
2. Cross-site scripting (XSS), incluindo DOM XSS
3. Injeção de SQL
4. Configuração incorreta do PHP
5. Ataques ao sistema de arquivos, incluindo inclusão de arquivos

Fault Injection

Procura apenas vulnerabilidades de injeção de falhas. Todas as verificações que não envolvem injeção são desativadas, e o Syhunt se concentra em injeção de SQL, XSS, inclusão de arquivos e falhas semelhantes.

Cross-Site Scripting (XSS)

Procura especificamente vulnerabilidades de XSS, incluindo XSS baseado em DOM.

Página 78 de 182

Syhunt Hybrid 7.2

SQL Injection

Procura especificamente vulnerabilidades de injeção de SQL e NoSQL.

File Inclusion

Procura especificamente vulnerabilidades de inclusão de arquivos e de travessia de diretórios (directory traversal).

Unvalidated Redirects

Procura especificamente vulnerabilidades de redirecionamento não validado.

Malware Content

Procura especificamente conteúdo malicioso, como:

- Backdoors web
- Conteúdo malicioso
- Parâmetros de depuração ocultos

Passive

Mapeia a estrutura do site e relata problemas sem realizar nenhum ataque, por exemplo:

- Vulnerabilidades em JavaScript do lado do cliente
- Fraquezas em formulários
- Divulgação de tecnologias web
- Cabeçalhos HTTP inseguros
- Software de servidor desatualizado e vulnerável
- Scripts referenciados desatualizados e vulneráveis
- Comentários HTML suspeitos
- Divulgação de código-fonte
- Conteúdo malicioso sendo servido

Spider Only

Mapeia a estrutura do site sem testar nem relatar nenhuma vulnerabilidade ou fraqueza.

Complete Scan

Testa todos os tipos de vulnerabilidades de aplicações web usando todas as mutações e técnicas de teste de invasão disponíveis, incluindo ataques de manipulação de cabeçalhos. Pode demorar bastante em servidores com muitas pastas e pontos de entrada.

Complete Scan, No DoS

Igual ao Complete Scan, mas com os testes de negação de serviço desativados.

Complete Scan, Paranoid

Igual ao Complete Scan, com uma força bruta de estrutura mais profunda. Pode demorar muito, principalmente em sites grandes. Também executa uma força bruta com verificação tripla para servidores que diferenciam maiúsculas de minúsculas, testando todas as variações de cada nome de arquivo (tudo em maiúsculas, tudo em minúsculas, iniciais maiúsculas e assim por diante).

Página 79 de 182

Syhunt Hybrid 7.2

Regras de Exclusão

> **Aplica-se a:** Syhunt Hybrid 7.2

As regras de exclusão (ignore rules) impedem que vulnerabilidades correspondentes sejam relatadas. Elas complementam os IDs de exclusão (exibidos ao final de cada vulnerabilidade no relatório) quando você precisa suprimir um conjunto mais amplo de resultados. Você pode adicioná-las:

- por alvo, na lista de exclusões do alvo (consulte Varredura Dinâmica);
- por repositório, em um arquivo `.vulnignore` (consulte Varredura de Código-Fonte);
- globalmente, para todas as varreduras (consulte Regras globais abaixo).

Sintaxe

Uma regra começa com `rule:` seguido de pares `parâmetro=valor` separados por vírgulas. Coloque o par entre aspas duplas se o valor contiver espaços ou vírgulas.

Exemplos

Regra	Efeito
`rule:path=*,checkid=C-1603659845-9557`	Ignora qualquer vulnerabilidade com o ID de verificação `C-1603659845-9557`.
`rule:path=*,name=XSS`	Ignora qualquer vulnerabilidade com "XSS" no título.
`rule:path=/demo/*,name=XSS`	Ignora vulnerabilidades com "XSS" no título cujo caminho começa com `/demo/`.
`rule:path=*,"name=Web Technology Disclosure"`	Ignora qualquer vulnerabilidade com "Web Technology Disclosure" no título.
`rule:path=/demo.php,lines=10`	*(SAST)* Ignora vulnerabilidades na linha 10 de `/demo.php`.
`rule:path=/demo.php,"lines=10,11"`	*(SAST)* Ignora vulnerabilidades nas linhas 10 e 11 de `/demo.php`.

Parâmetros

Parâmetro	Comparado com
`path` **(obrigatório)**	O caminho afetado. Aceita os curingas `?` e `*`.
`checkid`	O ID de verificação da vulnerabilidade.
`name`	O título da vulnerabilidade.
`params`	O(s) parâmetro(s) afetado(s). Separe vários parâmetros com vírgulas.
`risk`	O risco da vulnerabilidade: `low`, `medium`, `high` ou `info`.
`module`	O módulo que encontrou a vulnerabilidade: `dyn` ou `code`. Se omitido, a regra vale para resultados do Dynamic e do Code.
`lines`	A(s) linha(s) de código-fonte afetada(s). Separe várias linhas com vírgulas.
`cve`	Um ID de CVE entre as referências CVE da vulnerabilidade.
`cwe`	Um número de CWE entre as referências CWE da vulnerabilidade.

Página 80 de 182

Syhunt Hybrid 7.2

Regras globais

Para criar regras que valham para todas as varreduras dinâmicas ou de código:

1. Abra a tela Connectors (**Menu** → **Connectors**) e adicione um conector do tipo **List** chamado `VULNIGNORE-DAST` (para varreduras dinâmicas) ou `VULNIGNORE-SAST` (para varreduras de código).
2. Clique no conector na lista para adicionar ou editar suas regras.

Página 81 de 182

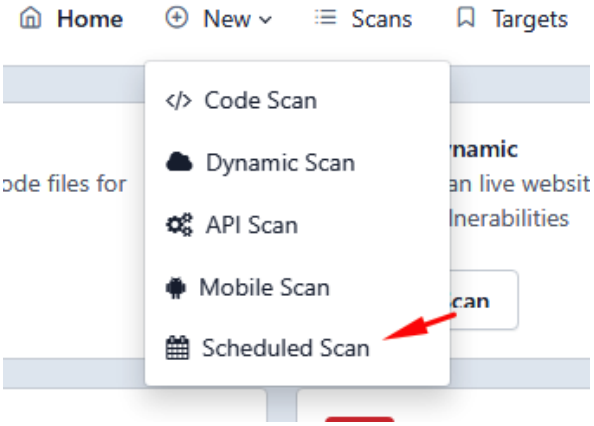
Syhunt Hybrid 7.2

Agendamento de Varreduras

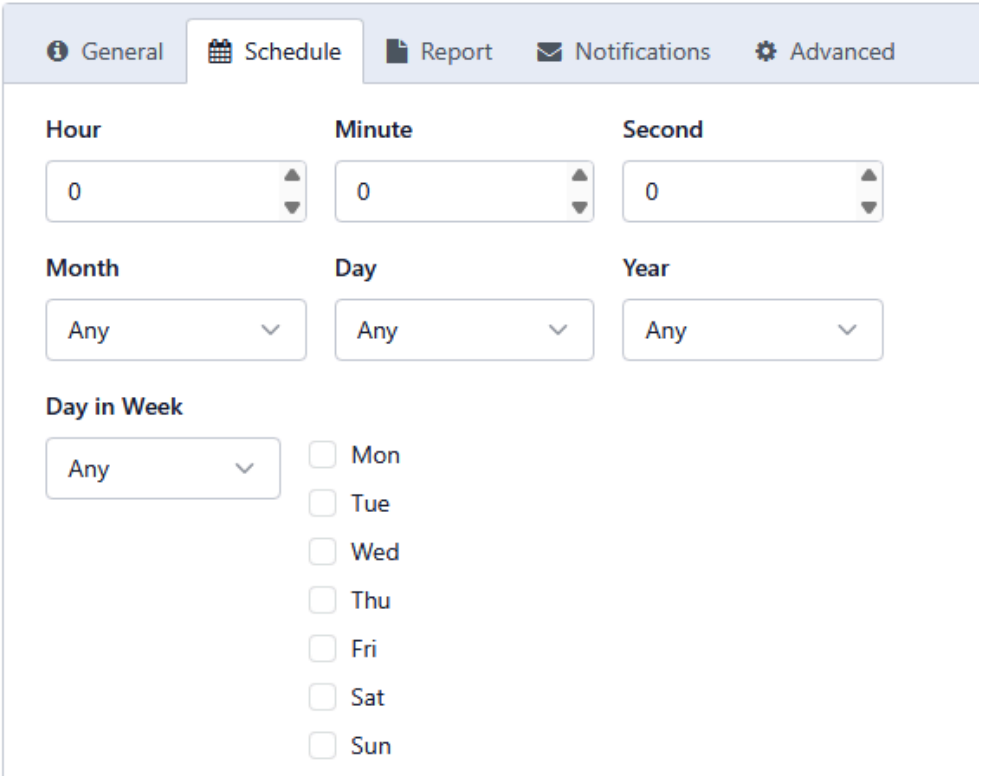
> **Aplica-se a:** Syhunt Hybrid 7.2

Como agendar uma varredura

1. Habilite a interface web, caso ainda não tenha feito isso.
2. Na interface web do Syhunt, clique em **New** → **Scheduled Scan**.



3. A tela **Schedule New** é aberta.
4. Na aba **General**, informe um nome para a varredura agendada (por exemplo, `MyScan`), preencha os dados do alvo e escolha um método de varredura.
5. Na aba **Report**, defina as opções de geração de relatório.
6. Na aba **Schedule**, defina quando a varredura deve ser executada.



7. Clique em **Save**.
8. Confirme que o novo agendamento aparece na lista de varreduras agendadas, em `https://127.0.0.1:8018/syhunt/?n=scanschedule`.

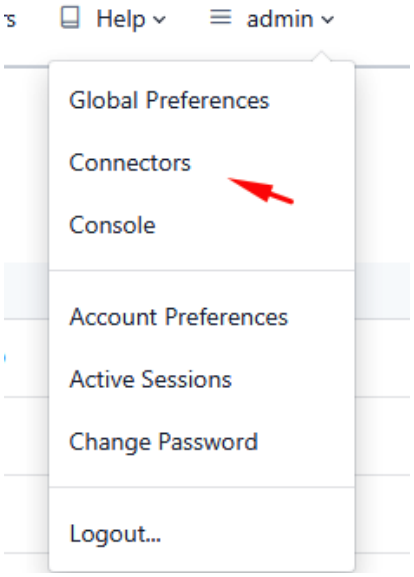
Página 82 de 182

Syhunt Hybrid 7.2

Envio de relatórios por e-mail

1. Crie um conector de e-mail

1. Clique em **Connectors** no menu superior da interface web.



2. Clique em **Add**. A tela **New Connector** é aberta.
3. Informe um nome para o conector (por exemplo, `MyEmail`).
4. Defina **Type** como **Email**.
5. Clique em **Create** e, após a mensagem de sucesso, em **Close**.

2. Configure o conector

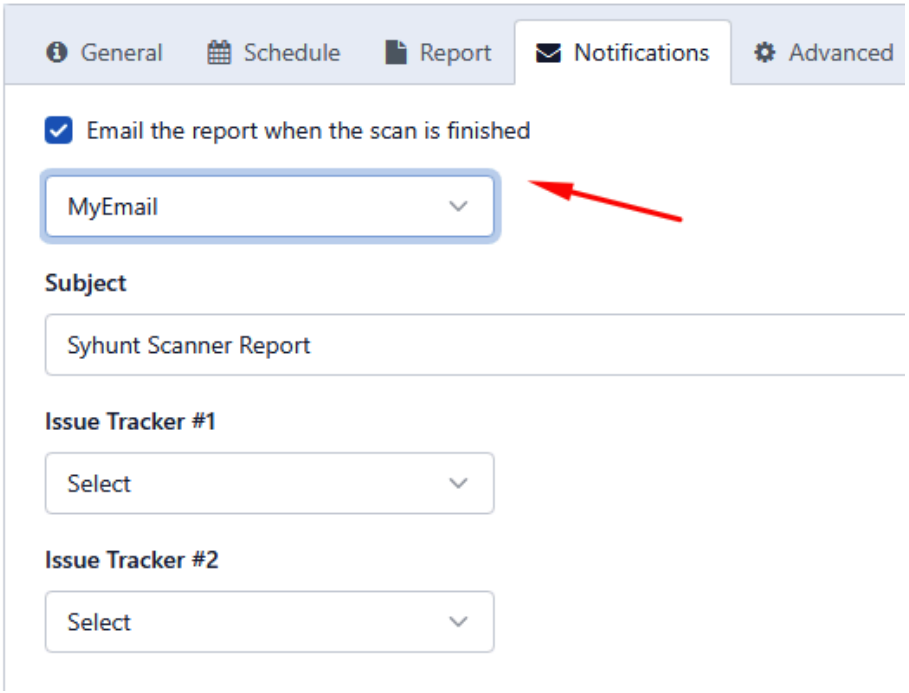
1. Clique no conector que você acabou de criar. A tela de preferências dele é aberta.
2. Informe os endereços de e-mail do remetente e do destinatário.
3. Informe o host SMTP e as credenciais de autenticação.
4. Clique em **Save**.

3. Associe o conector a uma varredura agendada

1. Abra **Scans** no menu superior e clique em **Scheduled Scans**.
2. Clique na varredura agendada cujos relatórios devem ser enviados por e-mail. A janela de preferências dela é aberta.

Página 83 de 182

3. Na aba **Notifications**, selecione **Email the report when the scan is finished**.



4. Escolha seu conector de e-mail na lista (`MyEmail` na captura de tela acima).
5. Clique em **Save**.

Consulta dos resultados

Você pode consultar os resultados de varreduras anteriores e em andamento a qualquer momento: abra a página inicial do Syhunt Hybrid e clique em **Scans** no menu superior.

Syhunt Hybrid 7.2

Análise Forense de Logs

> **Aplica-se a:** Syhunt Forensic 7.0.13

Importante
No Syhunt Hybrid 7.2, o Forensic está disponível apenas pela linha de comando (o comando `scanlog` ; consulte o guia de Linha de Comando). Uma atualização futura o tornará disponível novamente pela interface web. Os passos de interface abaixo se aplicam a versões anteriores.

Introdução

O Syhunt Forensic (antigo Syhunt Insight) investiga comprometimentos de aplicações web e ajuda a evitar novas invasões. Ele identifica com rapidez e precisão a origem, o tipo e os métodos de um ataque, para que as organizações possam corrigir as falhas e se defender de novas ameaças.

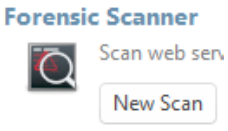
O Syhunt Forensic não é um produto independente. Ele complementa as ferramentas de teste dinâmico (DAST) e estático (SAST) do Syhunt, concentrando-se na análise pós-incidente e na investigação de invasões: enquanto o DAST e o SAST encontram e corrigem vulnerabilidades durante o desenvolvimento e a implantação, o Forensic mostra em detalhes o que aconteceu em ataques que já ocorreram. Juntos, eles cobrem tanto a prevenção quanto a resposta.

Formatos de log suportados

O Syhunt Forensic reconstrói sessões de ataque a partir dos logs dos servidores web mais populares. Ele se baseia nos detalhes registrados nesses logs para distinguir o tráfego legítimo do malicioso e os ataques automatizados dos manuais, por isso precisa ter acesso a eles:

- Apache
- Nginx
- Microsoft IIS
- Lighttpd

Fazendo uma varredura forense

1. Abra o Syhunt Hybrid e clique no ícone do Syhunt Forensic, ou em **New Scan**, na página de boas-vindas.


2. Selecione o arquivo de log do servidor web a ser analisado. Normalmente é um arquivo com extensão `.log` , como `access.log` , ou um arquivo cujo nome começa com `access_log` e não tem extensão.
3. Aguarde o fim da varredura.

Ao final da varredura, o log recebe um destes status:

Status	Significado
No attacks	Nenhum ataque à segurança de aplicações foi encontrado no log.
Attack Attempts Found	Há sinais de possíveis ataques, mas nenhuma invasão confirmada.
Breached	Há sinais de possíveis ataques **e** de uma invasão bem-sucedida.

Página 85 de 182

Syhunt Hybrid 7.2

Clique em **Save Results** na barra de ferramentas para salvar a lista de tentativas de ataque.

Reconstruindo a sessão de um atacante

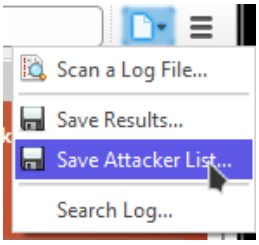
O Syhunt Forensic pode reconstruir sessões de ataque inteiras a partir do log, separando o tráfego legítimo do malicioso e os ataques automatizados dos manuais. Isso ajuda a entender a dimensão completa de um incidente.

1. Clique duas vezes em um ataque na lista.
2. Na barra inferior, clique em **View All Requests**.
3. Uma nova aba é aberta. Aguarde a conclusão da reconstrução.

Exportando a lista de atacantes

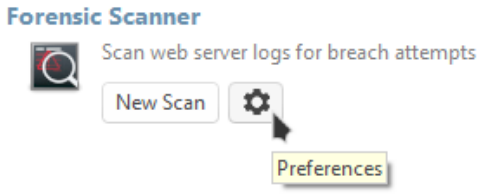
Você pode exportar os endereços IP identificados como atacantes e usar a lista para reforçar as regras do seu firewall, bloqueando esses endereços de forma proativa para evitar novos ataques das mesmas origens.

Ao final da varredura, clique no ícone de arquivo mostrado abaixo e escolha **Save Attacker List**.



Excluindo endereços IP da análise

1. Clique em **Preferences**.



2. Vá para a aba **Exclusions**.
3. Clique em **Source IPs**.
4. Adicione os endereços IP a serem ignorados ou importe-os de um arquivo.
5. Clique em **OK** para salvar a lista e depois em **OK** novamente para fechar a janela de preferências.
6. Execute a varredura novamente.

Página 86 de 182

Syhunt Hybrid 7.2

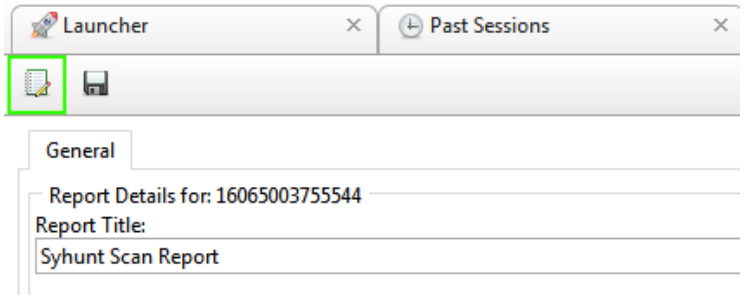
Relatórios

> **Aplica-se a:** Syhunt Hybrid 6.9

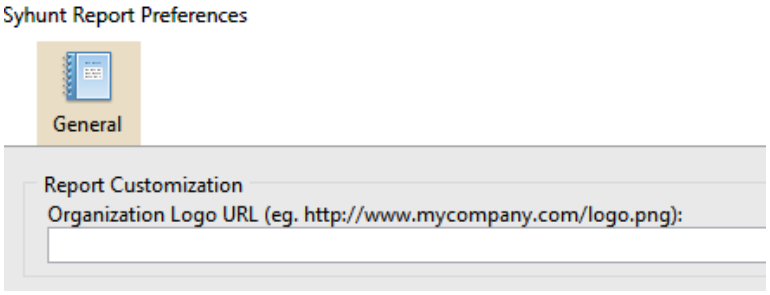
Personalização dos relatórios

Antes de salvar um relatório, você pode escolher o idioma e adicionar o logotipo da sua organização. As configurações valem para todos os relatórios gerados a partir de então.

1. Clique em **Edit Report Preferences** na barra de ferramentas. A janela **Report Preferences** é aberta.



2. Informe a URL da imagem que contém o seu logotipo.



3. Escolha o idioma de saída.
4. Clique em **OK** para salvar.

Os relatórios gerados a partir de agora exibirão o logotipo da sua organização no lugar do logotipo da Syhunt.

Exportação dos resultados

Ao final de uma varredura, clique em **Export** na tela de resultados para baixá-los no formato de sua preferência. Para a lista completa de formatos de relatório e exportação, consulte as Especificações Técnicas.

Página 87 de 182

Syhunt Hybrid 7.2

Integrações

Página 88 de 182

Syhunt Hybrid 7.2

Linha de Comando (CLI)

> **Aplica-se a:** Syhunt Hybrid 7.2

Conteúdo

- Introdução
- Varreduras dinâmicas
- Varreduras de API
- Varreduras de código-fonte
- Varreduras forenses
- Condições de aprovação/reprovação
- Modos de varredura incremental
- Formatos, modelos e idiomas de relatório
- Alterando as preferências do Syhunt
- Usando rastreadores de issues
- Atualizando o Syhunt

Introdução

Este guia explica como executar varreduras e gerar relatórios de vulnerabilidades pela linha de comando.

As ferramentas de linha de comando ficam na pasta de instalação do Syhunt, que depende do sistema operacional e da edição:

Sistema	Local padrão

Windows	`C:\Program Files\Syhunt Hybrid\`
	`C:\Program Files (x86)\Syhunt Hybrid\`
	`C:\Program Files\Syhunt Community`

Linux	`/home/<usuário>/syhunt-hybrid/carbon/` `/home/<usuário>/syhunt-community/carbon/`
macOS	`/Applications/Syhunt Hybrid/carbon` `/Applications/Syhunt Community/carbon`

Ferramenta	Finalidade
`scanurl`	Varreduras dinâmicas (DAST) e de API
`scancode`	Varreduras de código-fonte (SAST/MAST)
`scanlog`	Análise forense de logs
`scancore`	Preferências, rastreadores e comandos de manutenção
`scanupdate`	Atualizações (Linux e macOS)

Abra um terminal ou Prompt de Comando na pasta de instalação antes de executar os comandos abaixo.

Varreduras dinâmicas

```
scanurl <urlinicial> -hm:<metodo>
```

Exemplo:

Página 89 de 182

Syhunt Hybrid 7.2

```
scanurl http://www.somehost.com -hm:appscan
```

O `scanurl` gera e salva um relatório automaticamente, a menos que você informe `-nr`.

Parâmetros do `scanurl`

Todos os parâmetros são opcionais.

Parâmetro	Descrição	Padrão
`-sn:<nome>`	Nome da sessão; deve ser único. Se omitido, um ID único é gerado.	ID gerado
`-hm:<nome>`	Método de varredura a ser usado.	`appscan`
`-emu:<modo>`	Modo de emulação de navegador: `chrome`, `edge`, `firefox`, `msie` ou `safari`.	`chrome`
`-srcdir:<pasta>`	Pasta do código-fonte do alvo para uma varredura híbrida (por exemplo, `"C:\www\docs\"` ou `/home/user/www/`).	
`-tk:<rastreador>`	Envia as vulnerabilidades a um rastreador após a varredura. Pode ser combinado com `-pfcond`.	
`-tk2:<rastreador>`, `-tk3:<rastreador>`	Rastreadores adicionais, como acima.	
`-nr`	Não gera relatório após a varredura.	
`-or`	Abre o relatório após gerá-lo.	
`-rout:<arquivo>`	Nome do arquivo de relatório; a extensão define o formato.	`Report_<sessão>.html`
`-rtpl:<nome>`	Modelo de relatório.	`Standard`
`-xout:<arquivo>`	Nome do arquivo de exportação; a extensão define o formato.	`Export_<sessão>.xml`
`-xout2:<arquivo>`	Um segundo arquivo de exportação.	`Export_<sessão>.xml`
`-pfcond:<condição>`	Condição de aprovação/reprovação a ser verificada.	
`-nv`	Saída menos detalhada (erros e informações básicas continuam sendo exibidos).	
`-inc:<modo>`	Modo de varredura incremental.	`targetpref`
`-inctag:<nome>`	Armazena os dados da varredura incremental sob uma tag.	
`-mnl:<n>`	Número máximo de links por servidor.	`10000`
`-mnr:<n>`	Número máximo de novas tentativas.	`2`
`-tmo:<ms>`	Tempo limite das requisições, em milissegundos.	`8000`
`-tml:<tempo>`	Tempo máximo da varredura, por exemplo `1d`, `3h`, `2h30m`, `50m`.	Sem limite
`-ver:<v>`	Versão do HTTP.	`1.1`
`-nofris`	Não segue automaticamente um redirecionamento para fora do domínio na URL inicial.	
`-nodos`	Desativa os testes de negação de serviço.	
`-nojs`	Desativa a emulação e execução de JavaScript.	
`-atype:<tipo>`	Tipo de autenticação: `Basic`, `Form` ou `Manual`.	

Página 90 de 182

Syhunt Hybrid 7.2

Parâmetro	Descrição	Padrão
`-auser:<usuário>`	Usuário para autenticação.	
`-apass:<senha>`	Senha para autenticação.	
`-about`	Exibe informações sobre a versão.	
`-help` (ou `/?`)	Lista os parâmetros disponíveis.	

Varredura de endereços IPv6

Coloque endereços IPv6 entre colchetes, por exemplo `http://[2001:4860:0:2001::68]/index.php`.

Varreduras de API

As varreduras de API também usam o `scanurl`. A única diferença é que você informa a URL da **especificação da API** em vez de uma URL inicial. Se informar a URL da própria API, o Syhunt executará uma varredura dinâmica tradicional, sem mapeamento da API, validação de esquema ou testes específicos de API.

Por exemplo, se a especificação estiver em `https://example.com/openapi.json` e descrever uma API em `https://api.example.com`:

```
scanurl https://example.com/openapi.json -hm:appscan   # correto: varredura de API
scanurl https://api.example.com                        # errado: executa uma varredura DAST tradicional
```

O `scanurl` aceita os mesmos parâmetros opcionais nas varreduras de API. Consulte Varredura de APIs Web para mais detalhes.

Varreduras de código-fonte

```
scancode <alvo> -hm:<metodo>
```

Exemplos:

```
:: Repositórios Git
scancode git://sub.domain.com/repo.git
scancode https://github.com/user/repo.git -rb:master

:: Caminhos locais
scancode /source/www/
scancode c:\source\www\
scancode c:\source\www\file.php
scancode c:\mobile\myapp.apk
scancode "c:\source code\www\"
```

Repositórios Azure DevOps / TFS:

```
scancode https://dev.azure.com/user/project
scancode https://myserver/tfs/project
scancode collection:https://dev.azure.com/user$/project
```

O `scancode` gera e salva um relatório automaticamente, a menos que você informe `-nr`.

Parâmetros do `scancode`

Todos os parâmetros são opcionais.

Página 91 de 182

Syhunt Hybrid 7.2

Parâmetro	Descrição	Padrão
`-sn:<nome>`	Nome da sessão; deve ser único. Se omitido, um ID único é gerado.	ID gerado
`-hm:<nome>`	Método de varredura a ser usado.	`appscan`
`-rb:<branch>`	Branch do repositório Git.	
`-tfsv:<versão>`	Versão do TFS.	`default`
`-tk:<rastreador>`	Envia as vulnerabilidades a um rastreador após a varredura. Pode ser combinado com `-pfcond`.	
`-tk2:<rastreador>`, `-tk3:<rastreador>`	Rastreadores adicionais, como acima.	
`-nr`	Não gera relatório após a varredura.	
`-or`	Abre o relatório após gerá-lo.	
`-rout:<arquivo>`	Nome do arquivo de relatório; a extensão define o formato.	`Report_<sessão>.html`
`-rtpl:<nome>`	Modelo de relatório.	`Standard`
`-xout:<arquivo>`	Nome do arquivo de exportação; a extensão define o formato.	`Export_<sessão>.xml`
`-xout2:<arquivo>`	Um segundo arquivo de exportação.	`Export_<sessão>.xml`
`-pfcond:<condição>`	Condição de aprovação/reprovação a ser verificada.	
`-nv`	Saída menos detalhada (erros e informações básicas continuam sendo exibidos).	
`-inc:<modo>`	Modo de varredura incremental.	`targetpref`
`-inctag:<nome>`	Armazena os dados da varredura incremental sob uma tag.	
`-excp:<caminhos>`	Caminhos a excluir da análise, por exemplo `-excp:/path/*,/path2/*`.	
`-refurl:<url>`	Uma URL associada ao código, apenas para referência.	
`-noifa`	Desativa a análise de filtragem de entrada.	
`-tml:<tempo>`	Tempo máximo da varredura, por exemplo `1d`, `3h`, `2h30m`, `50m`.	Sem limite
`-about`	Exibe informações sobre a versão.	
`-help` (ou `/?`)	Lista os parâmetros disponíveis.	

Varreduras forenses

A análise forense de logs é executada com o `scanlog`. No Syhunt Hybrid 7.2, esta é a única forma de executar o Syhunt Forensic. Consulte Análise Forense de Logs. Execute `scanlog -help` para listar seus parâmetros.

Condições de aprovação/reprovação

Informe uma condição de aprovação/reprovação ao `scanurl` ou ao `scancode` com o parâmetro `-pfcond`.

Condição	Reprova se...
`high` ou `fail-if:risk=high`	for encontrada uma vulnerabilidade ou ameaça de risco Alto.

Página 92 de 182

Syhunt Hybrid 7.2

Condição	Reprova se...
`medium` ou `fail-if:risk=mediumup`	for encontrada uma vulnerabilidade ou ameaça de risco Médio ou Alto.
`low` ou `fail-if:risk=lowup`	for encontrada uma vulnerabilidade ou ameaça de risco Baixo, Médio ou Alto.

Modos de varredura incremental

Modo	Descrição
`auto`	Gerencia o cache de varredura incremental automaticamente (recomendado).
`disabled`	Desativa o cache de varredura incremental. As varreduras levam de 3 a 4 vezes mais tempo.
`forced`	Usa sempre a varredura incremental. Se usar este modo, execute também uma varredura normal, não forçada, cerca de uma vez por mês.
`targetpref`	Usa o modo definido nas preferências do alvo.

Para limpar o cache de varredura incremental, execute `scancore -runcmd:clearinc`.

Formatos, modelos e idiomas de relatório

Idiomas

Código	Idioma
`EN`	Inglês
`DE`	Alemão
`ES`	Espanhol
`FR`	Francês
`IT`	Italiano
`JP`	Japonês
`KR`	Coreano
`PT`	Português

Para definir o idioma dos relatórios:

```
scancore -prefset:hybrid.report.defaultlanguage -v:PT
```

Também é possível defini-lo pela interface: ☰ **Menu** → **Preferences** → **Other Hybrid Preferences**.

Formatos

O formato de saída é definido pela extensão do arquivo informado em `-rout` ou `-xout`:

Extensão	Formato
`.html`	Relatório HTML
`.pdf`	Relatório PDF
`.xml`	Exportação XML

Página 93 de 182

Syhunt Hybrid 7.2

Extensão	Formato
`.json`	Exportação de vulnerabilidades em JSON
`.csv`	CSV (valores separados por vírgula)
`.mse.csv`	CSV para Microsoft Excel
`.txt`	Relatório em texto

Modelos

Modelo	Conteúdo
`Standard`	Informações sobre vulnerabilidades de risco baixo a alto.
`Comparison`	O conteúdo padrão, mais a evolução das vulnerabilidades entre varreduras.
`Compliance`	Informações de conformidade com OWASP Top 10, CWE/SANS Top 25 (2019) e PCI DSS 3.2.1.
`Mobile`	Informações de conformidade com OWASP Mobile Top 10, CWE/SANS Top 25 (2019) e PCI DSS 3.2.1.
`Complete`	O conteúdo padrão, mais detalhes de comparação, conformidade, requisição/resposta e cobertura.

Para os métodos de varredura disponíveis e seus nomes na CLI, consulte Métodos de Varredura.

Alterando as preferências do Syhunt

Desde o Syhunt 6.9.6.3, você pode alterar as preferências persistentes do Syhunt, globalmente ou para um alvo específico, com `scancore -prefset`. Use `-prefprint` para exibir o valor atual de qualquer preferência.

```
-- Exemplo 1: definir o logotipo exibido nos novos relatórios
scancore -prefset:hybrid.report.company.logo.urldark -v:https://www.mydomain.com/mylogo.png

-- Exemplo 2: exibir a URL atual do logotipo dos relatórios
scancore -prefprint:hybrid.report.company.logo.urldark

-- Exemplo 3: definir autenticação Basic para um site específico
scancore -tg:http://127.0.0.1 -prefset:dynamic.formauth.type -v:Basic
scancore -tg:http://127.0.0.1 -prefset:dynamic.formauth.username -v:myuser
scancore -tg:http://127.0.0.1 -prefset:dynamic.formauth.password -vsecret

-- Exemplo 4: habilitar ou desabilitar as preferências de um site específico
scancore -tg:http://127.0.0.1 -prefset:enabled -v:true
scancore -tg:http://127.0.0.1 -prefset:enabled -v:false

-- Exemplo 5: definir um valor a partir de um arquivo de lista
scancore -tg:http://127.0.0.1 -prefset:dynamic.lists.cookies -fromfile:mycookies.lst
```

Consulte Preferências para a lista completa de IDs de preferências.

Limpando todas as preferências

```
-- Limpar as preferências globais
scancore -runcmd:clearpref
-- Limpar as preferências de todos os sites
scancore -runcmd:clearsite
```

Usando rastreadores de issues

Desde o Syhunt 6.9.6.3, você pode adicionar, gerenciar e usar rastreadores de issues pela linha de comando com `scancore -tracker`.

Página 94 de 182

Syhunt Hybrid 7.2

```
-- Exemplo 1: adicionar um rastreador do GitHub
scancore -tracker:add
-- Informe o tipo (GitHub) e um nome para o rastreador e pressione Enter.
-- Configure-o:
scancore -tracker:set to:mytrackername -key:project.name -v:owner/repo
scancore -tracker:set to:mytrackername -key:auth.token.encrypted -vsecret

-- Exemplo 2: adicionar um rastreador de e-mail
scancore -tracker:add
-- Informe o tipo (Email) e um nome para o rastreador e pressione Enter.
scancore -tracker:set to:myemailtracker -key:message.from -v:robot@yourdomain.com
scancore -tracker:set to:myemailtracker -key:message.tolist -v:security@yourdomain.com,team@yourdomain.com
scancore -tracker:set to:myemailtracker -key:auth.targethost -v:smtp.yourdomain.com
scancore -tracker:set to:myemailtracker -key:smtp.targetport -v:587
scancore -tracker:set to:myemailtracker -key:auth.username -v:myusername
scancore -tracker:set to:myemailtracker -key:auth.password -vsecret

-- Exemplo 3: testar um rastreador
scancore -tracker:send -tid:TEST -to:mytrackername -note:"Meu comentário"

-- Exemplo 4: enviar uma vulnerabilidade de um relatório a um rastreador
-- 1596281007-7-4771 é o track ID da vulnerabilidade, obtido no relatório.
scancore -tracker:send -tid:1596281007-7-4771 -to:mytrackername -note:"Meu comentário"

-- Exemplo 5: listar os rastreadores
scancore -tracker:list

-- Exemplo 6: excluir um rastreador
scancore -tracker:del to:mytrackername
```

Consulte Preferências: rastreadores para a lista de IDs de preferências de rastreadores.

Enviando vulnerabilidades a um rastreador após a varredura

Informe `-tk` ao `scanurl` ou ao `scancode` para enviar a um rastreador um resumo das vulnerabilidades encontradas:

```
-- Exemplo 1: usar um rastreador existente
scanurl mydomain -tk:trackername

-- Exemplo 2: enviar apenas se forem encontradas vulnerabilidades de severidade alta
scanurl mydomain -tk:trackername -pfcond:fail-if:risk=high

-- Exemplo 3: definir o rastreador na hora
scanurl mydomain -tk:"?app=github###project=user/repo###token=ghp_etc"
```

> **Observação**
> Ao informar preferências de rastreadores em `-tk`, use os nomes abreviados.

Removendo todos os rastreadores

```
scancore -runcmd:cleartrack
```

Atualizando o Syhunt

No Linux e no macOS, execute `scanupdate` para verificar se há atualizações; se houver, o Syhunt pergunta se deve baixá-las e instalá-las. Execute `scanupdate auto` para instalá-las sem pedir confirmação.

No Windows, baixe e instale as atualizações a partir do site da Syhunt.

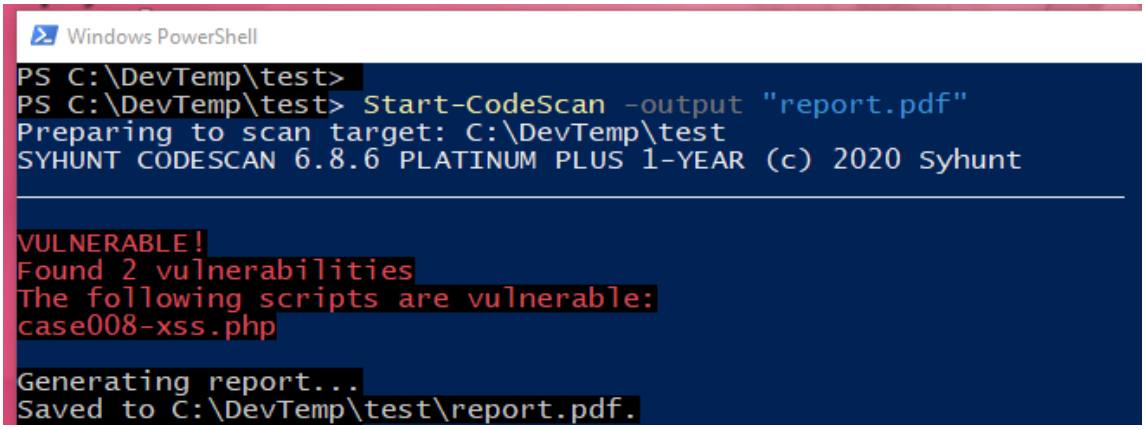
Página 95 de 182

Syhunt Hybrid 7.2

PowerShell

> **Aplica-se a:** Syhunt Hybrid 6.9.17

Você pode executar varreduras do Syhunt a partir de scripts PowerShell, o que facilita incluir o Syhunt Dynamic, o Syhunt Code e o Syhunt Mobile no seu pipeline de entrega contínua e em outras automações. O instalador do Syhunt Hybrid adiciona um **módulo PowerShell do Syhunt** no Windows, com funções para iniciar varreduras, obter relatórios e resultados e realizar testes de aprovação/reprovação.



Adicionando o Syhunt a um script

Insira um código como este no ponto adequado do seu script PowerShell:

```
# SAST: varredura da pasta/repositório atual
Start-CodeScan -pfcond "medium"

# SAST: varredura de um repositório remoto
$MyProject = @{
  target = 'https://github.com/syhunt/vulnphp.git';
  branch = 'main';
  pfcond = 'medium';
  output = 'report.pdf'
}
Start-CodeScan @MyProject

# DAST: varredura de uma URL
$MyWebsite = @{
  target = 'https://www.somewebsite.com';
  pfcond = 'medium';
  output = 'report.pdf'
}
Start-DynamicScan @MyWebsite
```

`Start-DynamicScan`

Inicia uma varredura do Syhunt Dynamic.

Parâmetro	Descrição
`target` **(obrigatório)**	A URL a ser verificada, por exemplo `http://www.somesite.com`.
`huntmethod`	O método de varredura a ser usado. Se omitido, é usado o método padrão.

Página 96 de 182

Syhunt Hybrid 7.2

Parâmetro	Descrição
`pfcond`	Faz o script terminar com código de falha quando uma condição é atendida. Consulte Condições de aprovação/reprovação.
`tracker`	O nome de um rastreador existente, ou um rastreador definido na hora, que receberá um resumo das vulnerabilidades encontradas. Consulte os exemplos.
`output`	Nome do arquivo de saída, por exemplo `report.pdf` ou `report.html`.
`outputex`	Nome de um segundo arquivo de saída, por exemplo `export.json`.
`verbmode`	`$false` por padrão. Defina como `$true` para uma saída detalhada (além de erros e informações básicas).
`genrep`	`$true` por padrão. Defina como `$false` para não gerar arquivo de saída.
`redirIO`	`$true` por padrão. Defina como `$false` para não redirecionar a entrada e a saída do scanner para o console.
`timelimit`	Tempo máximo da varredura, por exemplo `1d`, `3h`, `2h30m`, `50m`. A varredura é interrompida ao atingir o limite. Sem limite por padrão.

`output` e `outputex` aceitam todos os formatos de saída suportados pelo Syhunt. Os arquivos são salvos na pasta de trabalho atual, a menos que você informe o caminho completo.

```
# Exemplo 1: em uma linha
Start-DynamicScan -target 'https://www.somewebsite.com' -pfcond 'fail-if:risk=mediumup'

# Exemplo 2: parâmetros em uma hashtable
$MyWebsite = @{
  target = 'https://www.somewebsite.com';
  pfcond = 'medium';
  output = 'report.pdf'
}
Start-DynamicScan @MyWebsite
```

`Start-CodeScan`

Inicia uma varredura do Syhunt Code. Todos os parâmetros são opcionais.

Parâmetro	Descrição
`target`	A URL de um repositório, ou uma pasta ou arquivo de código-fonte local. Se omitido, é feita a varredura da pasta de trabalho atual.
`branch`	O branch a ser verificado. Se omitido, o cliente Git obtém o branch padrão.
`huntmethod`	O método de varredura a ser usado. Se omitido, é usado o método padrão.
`pfcond`	Faz o script terminar com código de falha quando uma condição é atendida. Consulte Condições de aprovação/reprovação.
`tracker`	O nome de um rastreador existente, ou um rastreador definido na hora, que receberá um resumo das vulnerabilidades encontradas. Consulte os exemplos.
`output`	Nome do arquivo de saída, por exemplo `report.pdf` ou `report.html`.
`outputex`	Nome de um segundo arquivo de saída, por exemplo `export.json`.
`verbmode`	`$false` por padrão. Defina como `$true` para uma saída detalhada.
`genrep`	`$true` por padrão. Defina como `$false` para não gerar arquivo de saída.
`redirIO`	`$true` por padrão. Defina como `$false` para não redirecionar a entrada e a saída do scanner para o console.
`timelimit`	Tempo máximo da varredura, por exemplo `1d`, `3h`, `2h30m`, `50m`. Sem limite por padrão.

Página 97 de 182

Syhunt Hybrid 7.2

```
# Exemplo 1: varredura da pasta/repositório atual
Start-CodeScan -pfcond "medium"

# Exemplo 2: varredura de um repositório Git remoto
Start-CodeScan -target "https://github.com/someuser/somerepo.git" -huntmethod "normal" -pfcond "medium"

# Exemplo 3: varredura de um projeto do Azure DevOps Services
Start-CodeScan -target "https://dev.azure.com/user/projectname" -huntmethod "normal" -pfcond "medium"

# Exemplo 4: varredura de uma pasta local específica
Start-CodeScan -target "C:\www\" -huntmethod "normal" -pfcond "medium"
```

Condições de aprovação/reprovação

Condição	Reprova se...
`high` ou `fail-if:risk=high`	for encontrada uma vulnerabilidade ou ameaça de risco Alto.
`medium` ou `fail-if:risk=mediumup`	for encontrada uma vulnerabilidade ou ameaça de risco Médio ou Alto.
`low` ou `fail-if:risk=lowup`	for encontrada uma vulnerabilidade ou ameaça de risco Baixo, Médio ou Alto.

Solução de problemas

"Script.ps1 cannot be loaded because running scripts is disabled on this system"

Abra o PowerShell como administrador e permita a execução de scripts:

```
Set-ExecutionPolicy RemoteSigned
```

"The term 'Start-CodeScan' is not recognized..."

O módulo do Syhunt não está carregado. Execute `Import-Module Syhunt` e chame a função novamente.

Carregando o módulo automaticamente

No Windows 10 e posteriores, o módulo do Syhunt é carregado automaticamente.

No Windows 7, execute `Import-Module Syhunt` antes de chamar as funções do Syhunt, ou adicione-o ao seu perfil do PowerShell para que seja carregado sempre:

1. Vá até a pasta do seu perfil do PowerShell. Por padrão, o perfil fica em:

   ```
   C:\Documents and Settings\User\My Documents\WindowsPowerShell\Microsoft.PowerShell_profile.ps1
   ```

2. Crie o arquivo `Microsoft.PowerShell_profile.ps1`.
3. Adicione a linha `Import-Module Syhunt` e salve o arquivo.

Página 98 de 182

Syhunt Hybrid 7.2

API Web

> **Aplica-se a:** Syhunt Hybrid 6.9.8.2

Conteúdo

- Introdução
- Iniciando uma varredura dinâmica
- Iniciando uma varredura de código-fonte
- Obtendo resultados

Introdução

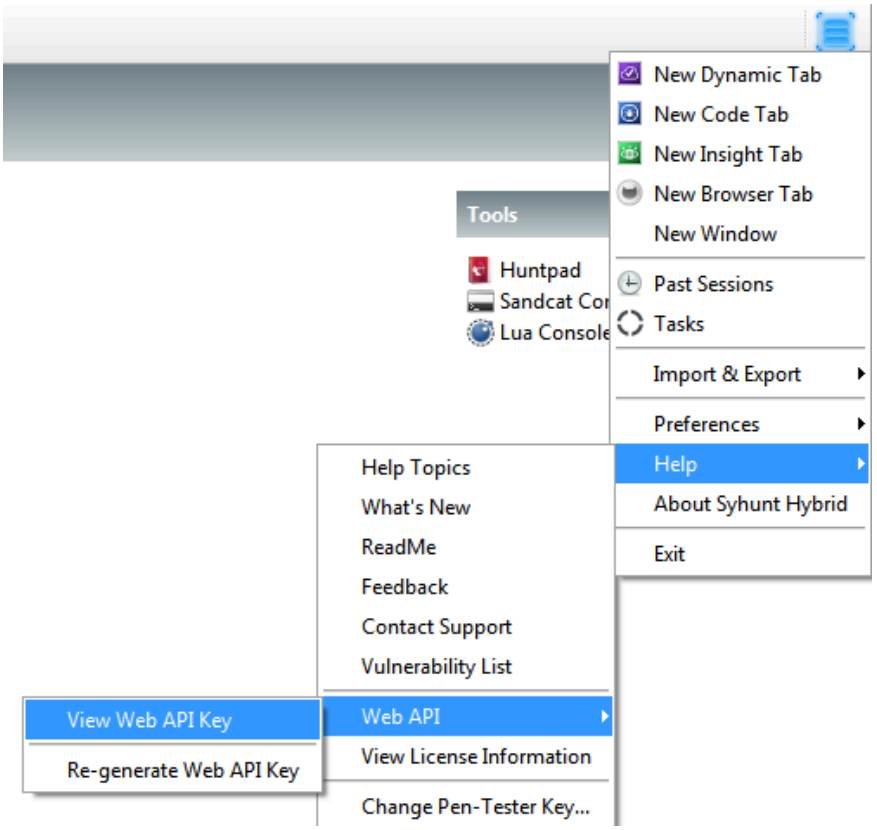
O Syhunt Hybrid Platinum inclui uma API web simples para iniciar varreduras dinâmicas e de código-fonte e obter o status, o relatório e o log de uma sessão de varredura. As requisições são do tipo `POST` com corpo JSON, e as respostas são em JSON, exceto quando você solicita um relatório, que é retornado em XML ou JSON.

As mesmas operações estão disponíveis em Lua; consulte a API Lua.

Antes de começar

Certifique-se de que o servidor da API do Syhunt está em execução e de que você tem uma chave de API.

- **Linux:** inicie o servidor da API com `scancore -apisignal:start` e gere uma chave de API com `scancore -apikeygen`.
- **Windows:** na pasta de instalação do Syhunt (normalmente `C:\Program Files\Syhunt Hybrid\`), abra a subpasta `Server` e clique duas vezes no executável `nginx` para iniciar o servidor. Para obter uma chave de API, abra o Syhunt Hybrid e escolha **Help** → **View Web API Key**.



Em seguida, abra `http://localhost:8017/` e verifique se aparece a página padrão do NGINX. A API fica disponível em:

Página 99 de 182

Syhunt Hybrid 7.2

- `http://<host>:8017/syhunt/launch.lua`
- `http://<host>:8017/syhunt/results.lua`

Toda requisição a esses endpoints deve incluir uma chave de API válida no corpo (`apikey`).

Iniciando uma varredura dinâmica

Endpoint: `POST /syhunt/launch.lua`

```
{
  "sessionname": "Test",
  "starturl": "http://127.0.0.1",
  "huntmethod": "appscan",
  "reporttemplate": "Complete",
  "apikey": "YOUR_API_KEY"
}
```

Chave	Descrição
`starturl` **(obrigatória)**	A URL alvo.
`apikey` **(obrigatória)**	Sua chave da API web do Syhunt (consulte Antes de começar).
`sessionname`	Um nome de sessão único e alfanumérico. Se omitido, um nome aleatório é gerado.
`huntmethod`	Um nome de método de varredura válido.
`reporttemplate`	Um modelo de relatório: `Standard` , `Comparison` , `Compliance` ou `Complete` .
`reportgl`	Se `true` , retorna saída JSON no formato do GitLab.
`timelimit`	Tempo máximo da varredura, por exemplo `1d` , `3h` , `2h30m` , `50m` . A varredura é interrompida ao atingir o limite. Sem limite por padrão.

Resposta:

```
{
  "huntmethod": "appscan",
  "pid": 5108,
  "result": true,
  "resultstr": "",
  "sessionname": "Test",
  "sessiontype": "dynamic"
}
```

Chave	Descrição
`pid`	Um ID de processo único da varredura em execução.
`result`	`true` se a varredura foi iniciada; `false` se não foi possível iniciá-la.
`resultstr`	Se `result` for `false` , uma descrição do erro.
`sessionname`	O nome da sessão da varredura. Se você não informou um, este é o nome gerado. Use-o nas requisições seguintes para obter os resultados.
`sessiontype`	O tipo da sessão: `dynamic` ou `code` .

Iniciando uma varredura de código-fonte

Endpoint: `POST /syhunt/launch.lua`

Página 100 de 182

Syhunt Hybrid 7.2

Pasta local:

```
{
  "sessionname": "Test",
  "sourcetarget": "P:\\Private\\MyWebApp\\",
  "huntmethod": "normal",
  "reporttemplate": "Complete",
  "apikey": "YOUR_API_KEY"
}
```

URL de repositório:

```
{
  "sessionname": "Test",
  "sourcetarget": "https://github.com/drnic/php-helloworld.git",
  "sourcebranch": "master",
  "huntmethod": "normal",
  "reporttemplate": "Complete",
  "apikey": "YOUR_API_KEY"
}
```

Chave	Descrição
`sourcetarget` **(obrigatória)**	Uma pasta local ou uma URL Git.
`apikey` **(obrigatória)**	Sua chave da API web do Syhunt.
`sourcebranch`	O branch a ser verificado, para URLs de repositório.
`sessionname`, `huntmethod`, `reporttemplate`, `reportgl`, `timelimit`	Como nas varreduras dinâmicas.

Resposta:

```
{
  "huntmethod": "normal",
  "pid": 4476,
  "result": true,
  "resultstr": "",
  "sessionname": "Test2",
  "sessiontype": "code"
}
```

As chaves da resposta são as mesmas das varreduras dinâmicas.

Obtendo resultados

Todas as requisições de resultados vão para **`POST /syhunt/results.lua`** e recebem:

Chave	Descrição
`sessionname` **(obrigatória)**	A sessão cujos resultados você quer obter.
`resulttype` **(obrigatória)**	`status`, `report_xml`, `report_json` ou `session_log`.
`apikey` **(obrigatória)**	Sua chave da API web do Syhunt.

Status

```
{
  "sessionname": "Test",
  "resulttype": "status",
  "apikey": "YOUR_API_KEY"
}
```


Página 101 de 182

Syhunt Hybrid 7.2

Resposta:

```
{
  "report_xml_available": true,
  "report_json_available": true,
  "session_log_available": true,
  "sessionname": "Test",
  "status": "Completed"
}
```

Chave	Descrição
`report_xml_available`	`true` se houver um relatório XML disponível.
`report_json_available`	`true` se houver um relatório JSON disponível.
`session_log_available`	`true` se houver um log de varredura disponível.
`status`	O status da sessão: `Scanning` ou, após o fim da varredura, `Completed`.

Relatório

Solicite `report_xml` ou `report_json`:

```
{
  "sessionname": "Test",
  "resulttype": "report_xml",
  "apikey": "YOUR_API_KEY"
}
```

Com `report_xml`, o corpo da resposta é o relatório da varredura em XML:

```
<?xml version="1.0"?>
<report>
  <report_title>Syhunt Scanner Report</report_title>
  <!-- ... -->
  <scanner_version>6.5.0.0</scanner_version>
</report>
```

Log

```
{
  "sessionname": "Test",
  "resulttype": "session_log",
  "apikey": "YOUR_API_KEY"
}
```

O corpo da resposta é o log da varredura, em texto simples.

Página 102 de 182

Syhunt Hybrid 7.2

API Lua

> **Aplica-se a:** Syhunt Hybrid 6.9.12

Conteúdo

- Introdução
- Iniciando uma varredura dinâmica
- Iniciando uma varredura de código-fonte
- Obtendo resultados
- Gerando um relatório ou exportação
- Obtendo detalhes da sessão
- Enviando um relatório por e-mail
- Atualizando preferências
- Outras funções

Introdução

O Syhunt Hybrid Platinum inclui uma API Lua simples para iniciar varreduras dinâmicas e de código-fonte, obter o status, o relatório e o log de uma sessão de varredura, e mais. As mesmas operações estão disponíveis via HTTP pela API Web.

Carregando a biblioteca do Syhunt

Carregue a biblioteca do Syhunt antes de usar a API:

```
-- Em um aplicativo com interface gráfica
require "SyMini"
-- Em um aplicativo de console
require "SyMini.Console"
```

Iniciando uma varredura dinâmica

```
local res = symini.scantask_launch({
  sessionname = "Test",
  starturl = "http://127.0.0.1",
  huntmethod = "appscan",
  reporttemplate = "Complete"
})
if res.result == true then
  print(res.sessionname)
end
```

Chave	Descrição
`starturl` **(obrigatória)**	A URL alvo.
`sessionname`	Um nome de sessão único e alfanumérico. Se omitido, um nome aleatório é gerado.
`huntmethod`	Um nome de método de varredura válido.
`reporttemplate`	Um modelo de relatório: `Standard`, `Comparison`, `Compliance` ou `Complete`.
`reportgl`	Se `true`, retorna saída JSON no formato do GitLab.
`timelimit`	Tempo máximo da varredura, por exemplo `1d`, `3h`, `2h30m`, `50m`. A varredura é interrompida ao atingir o limite. Sem limite por padrão.

Página 103 de 182

Syhunt Hybrid 7.2

Tabela retornada:

Chave	Descrição
`pid`	Um ID de processo único da varredura em execução.
`result`	`true` se a varredura foi iniciada; `false` se não foi possível iniciá-la.
`resultstr`	Se `result` for `false`, uma descrição do erro.
`sessionname`	O nome da sessão da varredura. Se você não informou um, este é o nome gerado. Use-o nas chamadas seguintes para obter os resultados.
`sessiontype`	O tipo da sessão: `dynamic` ou `code`.

Iniciando uma varredura de código-fonte

```
-- Varredura de uma pasta local
local res = symini.scantask_launch({
  sessionname = "Test",
  sourcetarget = "P:\\Private\\MyWebApp\\",
  huntmethod = "normal",
  reporttemplate = "Complete"
})

-- Varredura de uma URL de repositório
local res = symini.scantask_launch({
  sessionname = "Test",
  sourcetarget = "https://github.com/drnic/php-helloworld.git",
  sourcebranch = "master",
  huntmethod = "normal",
  reporttemplate = "Complete"
})

if res.result == true then
  print(res.sessionname)
end
```

Chave	Descrição
`sourcetarget` **(obrigatória)**	Uma pasta local ou uma URL Git.
`sourcebranch`	O branch a ser verificado, para URLs de repositório.
`sessionname`, `huntmethod`, `reporttemplate`, `reportgl`, `timelimit`	Como nas varreduras dinâmicas.

A tabela retornada é a mesma das varreduras dinâmicas.

Obtendo resultados

Status

```
local res = symini.scantask_getresults({
  sessionname = "Test",
  resulttype = "status"
})
```

Chave	Descrição
`sessionname` **(obrigatória)**	A sessão cujos resultados você quer obter.
`resulttype` **(obrigatória)**	`status`, `report_xml`, `report_json` ou `session_log`.

Página 104 de 182

Syhunt Hybrid 7.2

Tabela retornada:

Chave	Descrição
`report_xml_available`	`true` se houver um relatório XML disponível.
`report_json_available`	`true` se houver um relatório JSON disponível.
`session_log_available`	`true` se houver um log de varredura disponível.
`status`	O status da sessão: `Scanning` ou, após o fim da varredura, `Completed`.

Relatório

```
-- Relatório XML
local res = symini.scantask_getresults({
  sessionname = "Test",
  resulttype = "report_xml"
})

-- Relatório JSON
local res = symini.scantask_getresults({
  sessionname = "Test",
  resulttype = "report_json"
})
```

Log

```
local res = symini.scantask_getresults({
  sessionname = "Test",
  resulttype = "session_log"
})
```

Retorna o log da varredura como texto.

Gerando um relatório ou exportação

```
local res = symini.genreport({
  sessionname = 'mysessionname',
  outfilename = 'myreport.pdf',
  template = 'Standard'
})
```

Chave	Descrição
`sessionname` **(obrigatória)**	A sessão do relatório.
`outfilename` **(obrigatória)**	O nome do arquivo de saída do relatório; a extensão define o formato.
`template`	O modelo de relatório. Padrão: `Standard`.
`passfailcond`	Uma condição de aprovação/reprovação.

Tabela retornada:

Chave	Descrição
`outfilename`	O caminho completo do relatório gerado.
`result`	`true` se o relatório foi gerado.
`resultstr`	Status da geração do relatório.

Página 105 de 182

Syhunt Hybrid 7.2

Chave	Descrição
`passfail_result`	`true` se a condição de aprovação/reprovação foi aprovada.
`passfail_status`	Status da condição de aprovação/reprovação.
`passfail_resultstr`	Motivo do resultado de aprovação/reprovação.

Obtendo detalhes da sessão

```
local res = symini.getsessiondetails('mysessionname')
```

Tabela retornada:

Chave	Descrição
`huntmethod`	Método de varredura.
`datetime`	Data e hora da varredura.
`duration`	Duração da varredura.
`targets`	Alvo(s) da varredura.
`targetdesc`	Descrição do alvo.
`sourcedir`	Pasta de código-fonte do alvo, se houver.
`targetfile`	Arquivo alvo, se houver.
`targeturl`	URL alvo, se houver.
`vulncount`	Número de vulnerabilidades.
`attackcount`	Número de ataques (varreduras de logs de servidores web).
`ports`	Portas do alvo, se houver.
`status`	Status da sessão.
`resultsdesc`	Descrição dos resultados.
`launcher`	Nome da ferramenta que iniciou a sessão.

Enviando um relatório por e-mail

```
local res = symini.emailreport({
  tracker = 'myemailtrackername',
  filename = 'myreport.pdf',
  subject = 'My Test Report'
})
```

Chave	Descrição
`tracker` **(obrigatória)**	O nome de um rastreador de e-mail já configurado.
`filename` **(obrigatória)**	Caminho completo do relatório a ser enviado.
`subject`	Assunto do e-mail. Padrão: `Syhunt Scanner Report`.

Tabela retornada:

Página 106 de 182

Syhunt Hybrid 7.2

Chave	Descrição
`result`	`true` se o e-mail foi enviado.
`resultstr`	Status, incluindo eventual mensagem de erro.
`resultdbg`	Log de depuração da operação.

Atualizando preferências

Função	Descrição
`symini.prefs_set(key, value [, targeturl])`	Define uma preferência persistente, global ou de um alvo.
`symini.prefs_get(key [, targeturl])`	Obtém uma preferência persistente, global ou de um alvo.

Consulte Preferências para os IDs de preferências disponíveis.

```
-- Atualizar uma preferência de site
local res = symini.prefs_set('enabled', true, 'http://127.0.0.1')
-- Atualizar uma preferência global
local res = symini.prefs_set('hybrid.report.company.logo.url', 'https://www.mydomain.com/mylogo.png')
if res.result == true then
  print('Preferência atualizada.')
end
```

Tabela retornada:

Chave	Descrição
`result`	`true` se a operação foi bem-sucedida.
`resultstr`	Status, incluindo eventual mensagem de erro.
`value`	O valor da preferência (somente em `prefs_get`).

Outras funções

Função	Descrição
`symini.info.version`	A versão do Syhunt.
`symini.getsessionname()`	Retorna um nome de sessão aleatório que pode ser usado para iniciar uma varredura.
`symini.getsessioncomparison(sessionname1, sessionname2)`	Retorna uma comparação entre duas sessões.
`symini.getsessionlist([maxdays])`	Retorna a lista de sessões anteriores.
`symini.runcmd(cmdname)`	Executa um comando (veja abaixo).

Comandos disponíveis para `symini.runcmd`:

Comando	Efeito
`stop`	Interrompe todos os processos em execução desta instalação do Syhunt.
`clearinc`	Limpa o cache de varredura incremental.
`clearpref`	Limpa as preferências globais.
`clearsite`	Limpa as preferências de sites.

Página 107 de 182

Comando	Efeito
`cleartrack`	Remove todos os rastreadores de issues.

Mais exemplos, incluindo uso baseado em objetos, estão no repositório scantools.

Syhunt Hybrid 7.2

GitHub

> **Aplica-se a:** Syhunt Hybrid 7.2

Conteúdo

- Introdução
- Ativando um runner do Syhunt
- Adicionando o Syhunt a um workflow
- Funções de varredura e condições de aprovação/reprovação
- Enviando vulnerabilidades ao GitHub Issues

Introdução

Você pode iniciar varreduras do Syhunt a partir de um workflow do GitHub Actions, incluindo o Syhunt Dynamic, o Syhunt Code e o Syhunt Mobile no seu pipeline de entrega contínua. Também pode configurar o GitHub como rastreador de issues no Syhunt, para que as vulnerabilidades sejam enviadas às Issues de um projeto.

O workflow a seguir faz a varredura do código-fonte do repositório, reprova o job se forem encontradas vulnerabilidades de risco médio ou alto e anexa um relatório de vulnerabilidades em PDF aos artefatos do job:

```
name: syhunt test
on:
  push:
    branches:
      - main
jobs:
  build:
    runs-on: [self-hosted, Windows, X64]
    steps:
      - uses: actions/checkout@v2
      - run: Start-CodeScan -pfcond 'medium' -output 'report.pdf'
      - name: 'Syhunt Report'
        uses: actions/upload-artifact@v2
        if: failure()
        with:
          name: syhunt-report
          path: report.pdf
          if-no-files-found: error
```

Ativando um runner do Syhunt

O runner do Syhunt é um runner auto-hospedado (self-hosted) do GitHub Actions que recebe solicitações de varredura, executa as varreduras e devolve os resultados ao GitHub.

> **Importante**
> Por segurança e desempenho, instale o runner em uma máquina virtual separada ou em uma máquina física dedicada.

1. Instale o Git for Windows com as configurações padrão.
2. Instale o Syhunt Hybrid (`syhunt-hybrid-<versão>.exe`) com as configurações padrão.
3. No GitHub, abra a página principal do repositório e:
    1. Clique em **Settings**.
    2. Na barra lateral esquerda, clique em **Actions** e depois em **Runners**.

Página 109 de 182

Syhunt Hybrid 7.2

3. Clique em **New self-hosted runner**.
4. Selecione o sistema operacional **Windows** e a arquitetura **x64**.

O GitHub exibe instruções para baixar o aplicativo do runner e instalá-lo na sua máquina.

4. Siga essas instruções para registrar o runner com o token do repositório. Ao executar `config.cmd`:

   1. Pressione **Enter** para adicionar o runner ao grupo Default.
   2. Informe um nome para o runner, por exemplo `syhunt`.
   3. Informe um rótulo para o runner, por exemplo `syhunt`.
   4. As mensagens **√ Runner successfully added** e **√ Runner connection is good** confirmam o sucesso.
   5. Pressione **Enter** para manter `_work` como pasta de trabalho.
   6. Escolha se o runner será executado como serviço do Windows.

5. Execute `run.cmd`. Você verá:

```
√ Connected to GitHub
2022-02-04 19:50:21Z: Listening for Jobs
```

O Syhunt já pode ser chamado a partir de workflows do GitHub Actions.

> **Observação**
> O GitHub remove automaticamente um runner auto-hospedado que fica mais de 30 dias sem se conectar ao GitHub Actions. Além disso, cada execução de workflow é limitada a 72 horas; execuções que atingem esse limite são canceladas.

Adicionando o Syhunt a um workflow

Se o seu repositório ainda não tiver um arquivo de workflow, crie um, por exemplo `.github/workflows/syhunt.yml`.

Exemplo de SAST: faz a varredura do código-fonte do repositório, reprova o job se forem encontradas vulnerabilidades de risco médio ou alto e anexa um relatório em PDF aos artefatos do job:

```yaml
name: syhunt test
on:
  push:
    branches:
      - main
jobs:
  build:
    runs-on: [self-hosted, Windows, X64]
    steps:
      - uses: actions/checkout@v2
      - run: Start-CodeScan -pfcond 'medium' -output 'report.pdf'
      - name: 'Syhunt Report'
        uses: actions/upload-artifact@v2
        if: failure()
        with:
          name: syhunt-report
          path: report.pdf
          if-no-files-found: error
```

Exemplo de DAST: faz a varredura da aplicação web em produção após a implantação, reprova o job se forem encontradas vulnerabilidades de risco médio ou alto e anexa um relatório em PDF aos artefatos do job:

```yaml
name: syhunt test
on:
  push:
    branches:
      - main
jobs:
  build:
```


Página 110 de 182

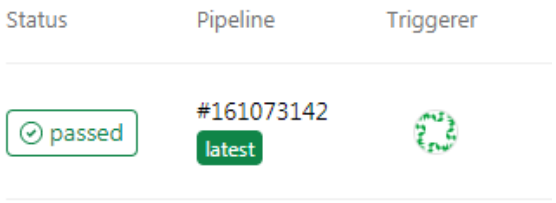
Syhunt Hybrid 7.2

```
    runs-on: [self-hosted, Windows, X64]
    steps:
      - uses: actions/checkout@v2
      - run: Start-DynamicScan -target 'www.productionurl.com' -pfcond 'medium' -output 'report.pdf'
      - name: 'Syhunt Report'
        uses: actions/upload-artifact@v2
        if: failure()
        with:
          name: syhunt-report
          path: report.pdf
          if-no-files-found: error
```

Funções de varredura e condições de aprovação/reprovação

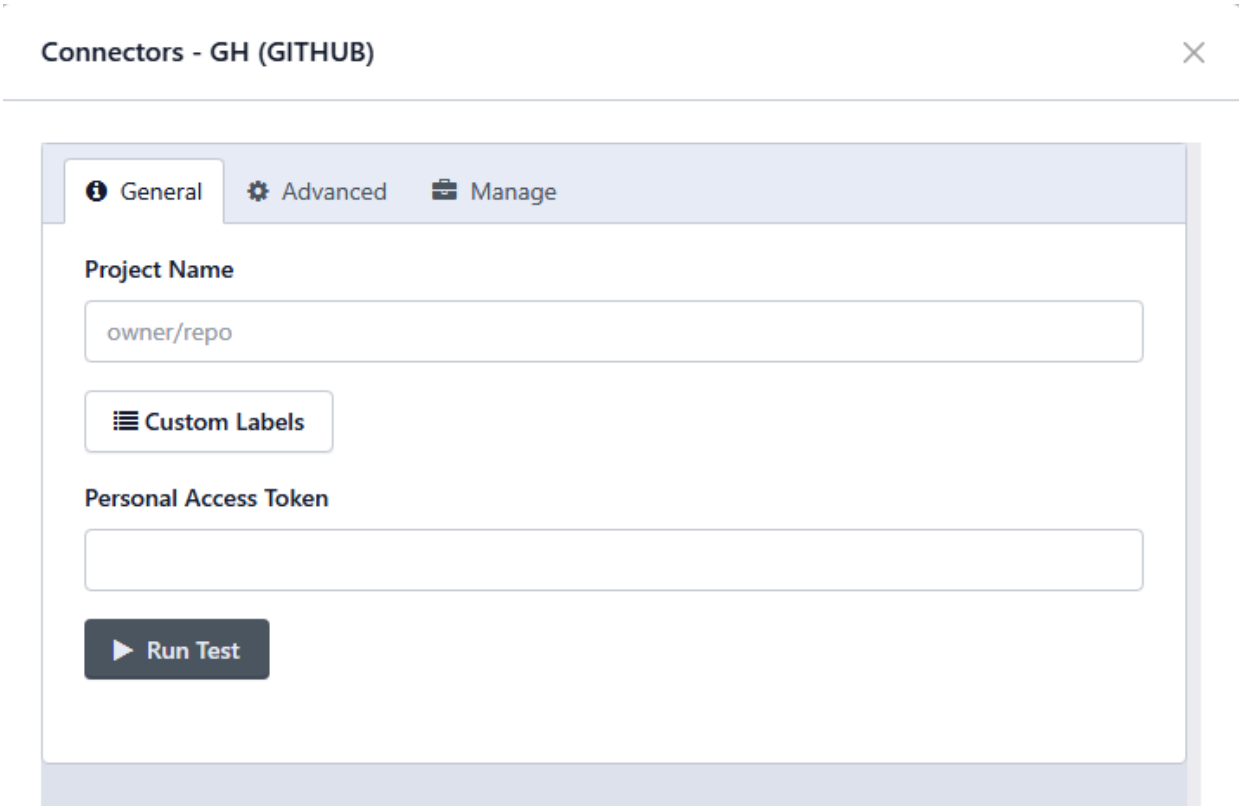
Os passos do workflow chamam as funções PowerShell do Syhunt. Consulte o guia de PowerShell para ver todos os parâmetros e mais exemplos:

- `Start-DynamicScan`
- `Start-CodeScan`
- Mais exemplos
- Condições de aprovação/reprovação



Enviando vulnerabilidades ao GitHub Issues

Com um rastreador de issues do GitHub configurado, você pode enviar uma vulnerabilidade a um projeto com um clique.



Página 111 de 182

Syhunt Hybrid 7.2

Consulte Rastreadores de Issues: GitHub para configurar o rastreador e Enviando vulnerabilidades a um rastreador para saber como enviar os resultados.

Página 112 de 182

Syhunt Hybrid 7.2

GitLab

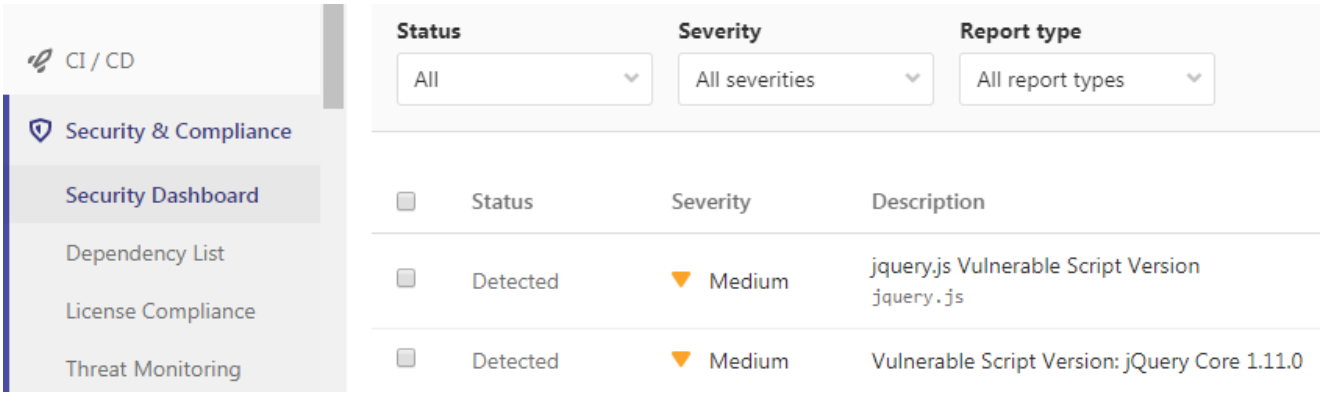
> **Aplica-se a:** Syhunt Hybrid 7.2

Conteúdo

- Introdução
- Ativando um runner do Syhunt
- Adicionando o Syhunt ao seu arquivo de CI
- Painel de segurança do GitLab
- Funções de varredura e condições de aprovação/reprovação
- Configurações avançadas do runner
- Enviando vulnerabilidades ao GitLab Issues

Introdução

Você pode iniciar varreduras do Syhunt a partir de um pipeline do GitLab CI, incluindo o Syhunt Dynamic, o Syhunt Code e o Syhunt Mobile no seu pipeline de entrega contínua e no painel de segurança, agendar varreduras e muito mais. Também pode configurar o GitLab como rastreador de issues no Syhunt, para que as vulnerabilidades sejam enviadas às Issues de um projeto.



O job a seguir faz a varredura do código-fonte do repositório, é reprovado se forem encontradas vulnerabilidades de risco médio ou alto, anexa um relatório em PDF aos artefatos do job e adiciona os resultados ao painel de segurança do GitLab:

```
syhunt_test:
  script:
    - Start-CodeScan -pfcond 'medium' -output 'report.pdf' -outputex 'gl-sast-report.json'
  artifacts:
    reports:
      sast: gl-sast-report.json
    paths:
      - report.pdf
    when: on_failure
  tags:
    - syhunt
```

Ativando um runner do Syhunt

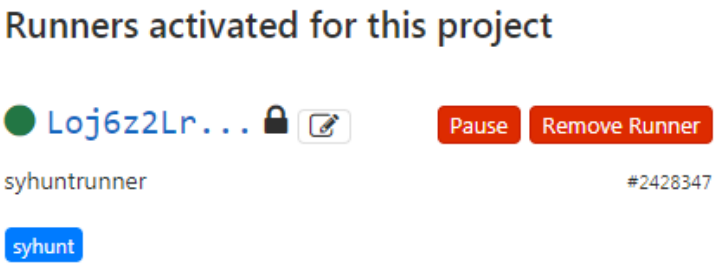
O runner do Syhunt é um serviço de CI que recebe solicitações de varredura, executa as varreduras e devolve os resultados ao GitLab.

Página 113 de 182

> **Importante**
> Por segurança e desempenho, instale o runner em uma máquina virtual separada ou em uma máquina física dedicada.

1. No seu projeto do GitLab, obtenha um token de registro de runner:
    1. Clique em **Settings** → **CI/CD**.
    2. Expanda **Runners**.
    3. Role até **Set up a specific Runner manually**.
    4. Copie o token de registro e guarde-o em local seguro; você vai precisar dele em seguida.
2. Instale o Git for Windows com as configurações padrão.
3. Instale o Syhunt Hybrid (`syhunt-hybrid-<versão>.exe`) com as configurações padrão.
4. Baixe e execute o instalador do Syhunt Runner. Quando forem solicitados os dados de registro do runner, cole o token de registro, clique em **Next** e conclua a instalação.

Volte à seção **Runners** no GitLab e atualize a página (F5). O `syhuntrunner` deve aparecer em **Runners activated for this project**:



O Syhunt já pode ser chamado a partir do seu pipeline de CI.

Adicionando o Syhunt ao seu arquivo de CI

Se o seu repositório ainda não tiver um arquivo de CI, vá em **Project overview** e clique em **Set up CI/CD**. Isso cria um arquivo `.gitlab-ci.yml` no repositório.

Exemplo de SAST: faz a varredura do código-fonte do repositório, é reprovado se forem encontradas vulnerabilidades de risco médio ou alto, anexa um relatório em PDF aos artefatos do job e adiciona os resultados ao painel de segurança:

```
syhunt_test:
  stage: test
  script:
    - Start-CodeScan -pfcond 'medium' -output 'report.pdf' -outputex 'gl-sast-report.json'
  artifacts:
    reports:
      sast: gl-sast-report.json
    paths:
      - report.pdf
    when: on_failure
  tags:
    - syhunt
```

Exemplo de DAST: faz a varredura da aplicação web em produção após a implantação, é reprovado se forem encontradas vulnerabilidades de risco médio ou alto, anexa um relatório em PDF aos artefatos do job e adiciona os resultados ao painel de segurança:

```
production:
  stage: deploy
  script:
    - Start-DynamicScan -target 'www.productionurl.com' -pfcond 'medium' -output 'report.pdf' -outputex 'gl-
dast-report.json'
  artifacts:
    reports:
```

Syhunt Hybrid 7.2

```
      dast: gl-dast-report.json
    paths:
      - report.pdf
    when: on_failure
  tags:
    - syhunt
  only:
    - tags
```

Para mais exemplos, consulte o guia de PowerShell.

Painel de segurança do GitLab

O Syhunt gera um relatório de vulnerabilidades compatível com o GitLab quando o nome do arquivo em `-outputex` é:

- **SAST:** `gl-sast-report.json` ou `<qualquernome>.gls.json`
- **DAST:** `gl-dast-report.json` ou `<qualquernome>.gld.json`

Para ativar a integração, adicione também o arquivo em `artifacts.reports.sast` ou `artifacts.reports.dast` no seu arquivo de CI, como nos exemplos acima.

Como o GitLab exibe os resultados

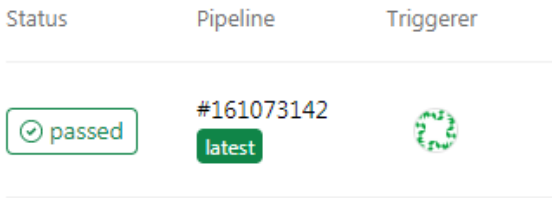
- Atualmente, o GitLab adiciona ao painel de segurança e ao relatório de vulnerabilidades de um projeto apenas os pipelines **bem-sucedidos**. Para ver todos os resultados ali, omita o `-pfcond` no passo do Syhunt. Como alternativa, com a condição de aprovação/reprovação `high`, se não forem encontradas vulnerabilidades de risco alto, todos os resultados de risco médio a informativo aparecem no painel.
- A aba **Security** do pipeline sempre lista todas as vulnerabilidades, seja o pipeline bem-sucedido ou não. Porém, se o pipeline falhar porque uma condição de aprovação/reprovação foi atendida, o GitLab pode mostrar, de forma inconsistente, uma contagem de 0 vulnerabilidades e ainda assim listá-las. Este é um problema conhecido do GitLab.

> **Dica**
> Se esta for a primeira varredura de um código **grande**, faça uma varredura sem a integração com o painel para ver quantas vulnerabilidades existem. Se houver muitas, corrija o primeiro lote relatado pelo Syhunt para melhorar a segurança da aplicação e só então ative a integração com o painel.

Funções de varredura e condições de aprovação/reprovação

Os jobs de CI chamam as funções PowerShell do Syhunt. Consulte o guia de PowerShell para ver todos os parâmetros:

- `Start-DynamicScan`
- `Start-CodeScan`
- Condições de aprovação/reprovação



Configurações avançadas do runner

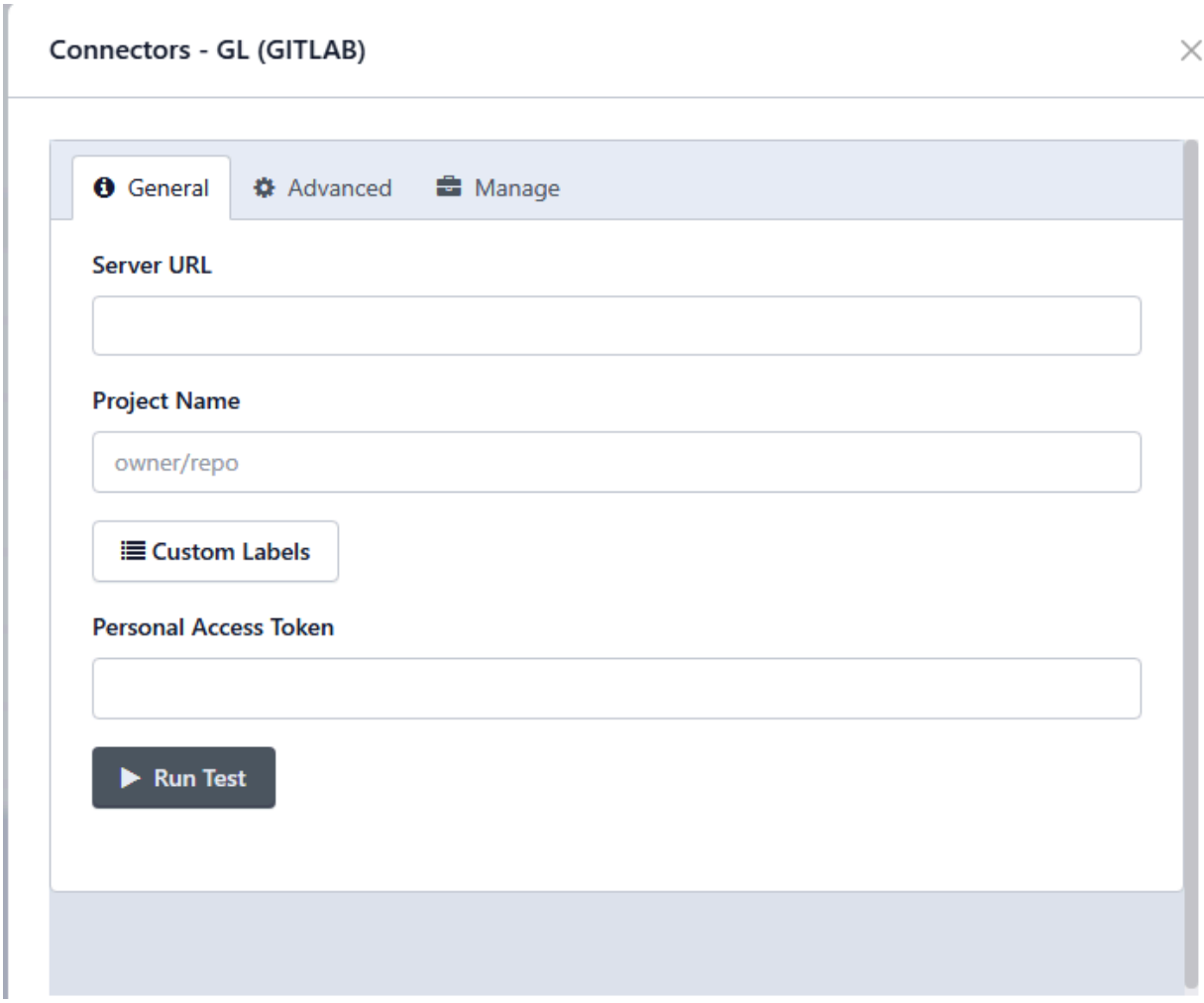
Para permitir que o runner processe vários jobs ao mesmo tempo, aumente o valor de `concurrent` em `C:\SyhuntRunner\config.toml`. Consulte a documentação de configuração avançada do GitLab.

Página 115 de 182

Syhunt Hybrid 7.2

Enviando vulnerabilidades ao GitLab Issues

Com um rastreador de issues do GitLab configurado, você pode enviar uma vulnerabilidade a um projeto com um clique.



Consulte Rastreadores de Issues: GitLab para configurar o rastreador e Enviando vulnerabilidades a um rastreador para saber como enviar os resultados.

Página 116 de 182

Syhunt Hybrid 7.2

Bitbucket

> **Aplica-se a:** Syhunt Hybrid 7.2

Conteúdo

- Introdução
- Ativando um runner do Syhunt
- Adicionando o Syhunt ao seu pipeline
- Funções de varredura e condições de aprovação/reprovação
- Bitbucket Issues

Introdução

Você pode iniciar varreduras do Syhunt a partir de um arquivo do Bitbucket Pipelines, incluindo o Syhunt Dynamic, o Syhunt Code e o Syhunt Mobile no seu pipeline de entrega contínua. Há suporte a runners no Windows e no Linux.

Os pipelines a seguir fazem a varredura do código-fonte do repositório, reprovam o passo se forem encontradas vulnerabilidades de risco médio ou alto e anexam um relatório de vulnerabilidades aos artefatos do passo.

Runner Windows:

```
pipelines:
  default:
    - step:
        runs-on:
          - self.hosted
          - windows
        script:
          - Start-CodeScan -pfcond 'fail-if:risk=mediumup' -output 'report.pdf'
        artifacts:
          - report.pdf
```

Runner Linux:

```
pipelines:
  default:
    - step:
        runs-on:
          - self.hosted
          - linux.shell
        script:
          - ${HOME}/syhunt-hybrid/carbon/scancode ${PWD} -pfcond:fail-if:risk=mediumup -
rout:${PWD}/report.html
        artifacts:
          - report.html
```

Ativando um runner do Syhunt

O runner do Syhunt é um runner auto-hospedado (self-hosted) do Bitbucket que recebe solicitações de varredura, executa as varreduras e devolve os resultados ao Bitbucket.

> **Importante**
> Por segurança e desempenho, instale o runner em uma máquina virtual separada ou em uma máquina física dedicada.

1. Instale o Git com as configurações padrão:

Página 117 de 182

Syhunt Hybrid 7.2

- **Windows:** Git for Windows.
- **Linux:** o pacote `git`, pelo gerenciador de pacotes da sua distribuição.

2. Instale o Syhunt Hybrid com as configurações padrão (`syhunt-hybrid-<versão>.exe` no Windows, `syhunt-hybrid-<versão>.jar` no Linux).

3. Execute `java -version` no Prompt de Comando ou no terminal e verifique se o Java 22.0.2 ou posterior está instalado. Caso contrário, instale o JDK 22 ou posterior.

4. No Bitbucket, abra o repositório e:

    1. Clique em **Repository settings** na barra lateral esquerda.
    2. Role até **Pipelines** e clique em **Runners**.
    3. Clique em **Add runner**. Em **System and architecture**, selecione **Windows (64-bit)**, ou **Linux Shell** se o Syhunt estiver instalado no Linux.
    4. Dê um nome ao runner (por exemplo, `Syhunt`) e um rótulo (por exemplo, `syhunt`) e clique em **Next**.

    O Bitbucket exibe instruções para baixar o aplicativo do runner e instalá-lo na sua máquina.

5. Siga essas instruções para ativar o runner e clique em **Next**.

O Syhunt já pode ser chamado a partir dos seus pipelines do Bitbucket.

Adicionando o Syhunt ao seu pipeline

Se o seu repositório ainda não tiver um arquivo de pipeline, crie `bitbucket-pipelines.yml` na raiz.

SAST

Faz a varredura do código-fonte do repositório, reprova o passo se forem encontradas vulnerabilidades de risco médio ou alto e anexa um relatório aos artefatos do passo.

Windows:

```
pipelines:
  default:
    - step:
        runs-on:
          - self.hosted
          - windows
        script:
          - Start-CodeScan -pfcond 'fail-if:risk=mediumup' -output 'report.pdf'
        artifacts:
          - report.pdf
```

Linux:

```
pipelines:
  default:
    - step:
        runs-on:
          - self.hosted
          - linux.shell
        script:
          - ${HOME}/syhunt-hybrid/carbon/scancode ${PWD} -pfcond:fail-if:risk=mediumup -
rout:${PWD}/report.html
        artifacts:
          - report.html
```

DAST

Faz a varredura da aplicação web em produção após a implantação, reprova o passo se forem encontradas vulnerabilidades de risco médio ou alto e anexa um relatório aos artefatos do passo.

Página 118 de 182

Syhunt Hybrid 7.2

Windows:

```
pipelines:
  default:
    - step:
        runs-on:
          - self.hosted
          - windows
        script:
          - Start-DynamicScan -target 'www.productionurl.com' -pfcond 'medium' -output 'report.pdf'
        artifacts:
          - report.pdf
```

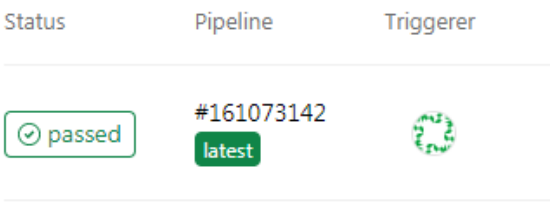
Linux:

```
pipelines:
  default:
    - step:
        runs-on:
          - self.hosted
          - linux.shell
        script:
          - ${HOME}/syhunt-hybrid/carbon/scanurl www.productionurl.com -pfcond:fail-if:risk=mediumup -
rout:${PWD}/report.html
        artifacts:
          - report.html
```

Funções de varredura e condições de aprovação/reprovação

	Runner Windows	**Runner Linux**
Varreduras dinâmicas	`Start-DynamicScan` (PowerShell)	`scanurl`
Varreduras de código	`Start-CodeScan` (PowerShell)	`scancode`
Condições de aprovação/reprovação	PowerShell	CLI

Mais exemplos para Windows estão no guia de PowerShell.



Bitbucket Issues

A integração com o Bitbucket Pipelines já está disponível. A integração com o Bitbucket Issues está prevista para uma versão futura.

Página 119 de 182

Syhunt Hybrid 7.2

Azure DevOps

> **Aplica-se a:** Syhunt Hybrid 7.2

Desde a versão 6.9.6, o Syhunt Community e o Syhunt Hybrid podem se conectar ao Azure DevOps Services, ao Azure DevOps Server e ao Team Foundation Server (TFS) e fazer a varredura do código-fonte dos projetos em busca de vulnerabilidades e fraquezas de segurança de aplicações.

Este recurso está disponível pela interface e pela CLI no Windows e funciona com projetos locais e hospedados na nuvem, tanto TFVC quanto Git. Sem configuração adicional, o Syhunt suporta o Azure DevOps Services, o Azure DevOps Server 2020 e 2019 e o TFS 2018 até o 2015. Como explicado abaixo, ele também pode ser configurado para se conectar a versões mais antigas do TFS (2013, 2012 e 2010).

Para automação de pipelines, consulte também a integração com PowerShell.

> **Observação**
> O suporte ao TFS no Syhunt Community e no Hybrid se baseia na ferramenta de exportação desenvolvida pelo Microsoft MVP Neno Loje, disponível gratuitamente no site dele. Ela acompanha o Syhunt com a permissão do desenvolvedor.

Versões suportadas

Versão	Implantação
Azure DevOps Services 2021	Nuvem
Azure DevOps Server 2020 / 2019	Local (on-premises)
Team Foundation Server 2018 / 2015	Local (on-premises)
Team Foundation Server 2013 / 2012 / 2010	Local (on-premises)*

* A varredura de projetos no TFS 2010–2013 exige o Microsoft Visual Studio ou o Microsoft Team Explorer na mesma máquina do Syhunt.

Para fazer a varredura de repositórios Git (e não apenas TFVC), instale o Git for Windows na mesma máquina.

Formatos de URL de projeto

Repositório	Exemplo de alvo
Azure DevOps Services, TFVC	`https://dev.azure.com/user/projectname`
Azure DevOps Services, Git	`https://dev.azure.com/user/projectname/_git/projectname`
Azure DevOps Server	`https://myserver/tfs/projectname`
Azure DevOps Services / Server / TFS, por coleção	`collection:https://dev.azure.com/user$/projectname`

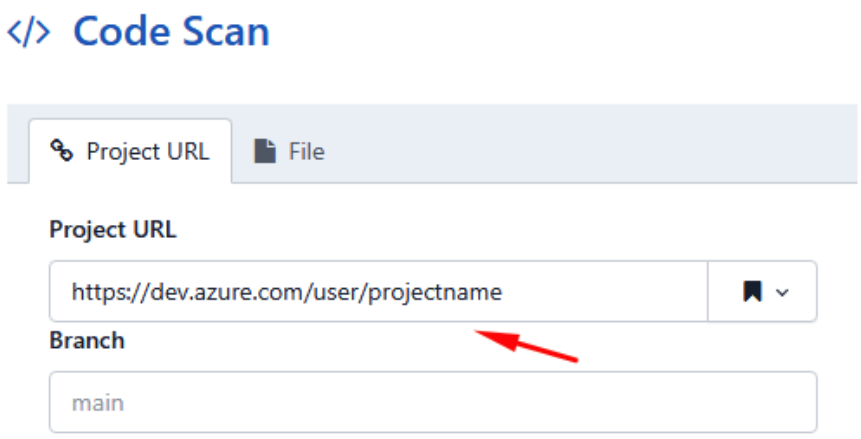
Com o formato `collection:`, o Syhunt se conecta à coleção de projetos em `https://dev.azure.com/user` e faz a varredura do caminho de servidor `$/projectname`.

Página 120 de 182

Syhunt Hybrid 7.2

Varredura pela interface

Na interface web do Syhunt Hybrid (ou na interface gráfica do Community), clique em **New Scan**, selecione **Project URL** e informe a URL do projeto, conforme descrito em Varredura de Código-Fonte. Isso funciona para projetos Azure DevOps e TFS/TFVC, bem como para repositórios Git hospedados no Azure DevOps. Você também pode agendar varreduras de código de URLs de projeto com o agendador de varreduras.



O Syhunt pede suas credenciais do Azure DevOps antes de se conectar, a menos que você já tenha se autenticado.

Para se conectar a um servidor TFS mais antigo (2010 a 2013), escolha a versão na aba **Advanced**. Como mencionado acima, isso exige o Visual Studio ou o Team Explorer na mesma máquina.

Varredura pela linha de comando

```
scancode https://dev.azure.com/user/projectname
scancode https://myserver/tfs/projectname
scancode collection:https://dev.azure.com/user$/projectname
```

O Syhunt pede suas credenciais do Azure DevOps antes de se conectar, a menos que você já tenha se autenticado.

Para servidores TFS mais antigos, informe a versão com `-tfsv`:

```
scancode https://myserver/tfs/projectname -tfsv:2013
scancode https://myserver/tfs/projectname -tfsv:2012
scancode https://myserver/tfs/projectname -tfsv:2010
```

Consulte o guia da CLI para mais exemplos.

Integração com pipelines

Você pode automatizar varreduras do Azure DevOps com:

- **PowerShell**: consulte o guia de PowerShell.
- **A CLI**: como mostrado acima e no guia da CLI.

Página 121 de 182

Syhunt Hybrid 7.2

Jenkins

> **Aplica-se a:** Syhunt Hybrid 7.2

Conteúdo

- Introdução
- Ativando um agente do Syhunt
- Antes de instalar o Syhunt (Windows)
- Antes de instalar o Syhunt (Linux)
- Adicionando o Syhunt a um pipeline
- Função `scanURL`
- Função `scanCode`
- Condições de aprovação/reprovação
- Agendamento de varreduras

Introdução

Você pode executar varreduras do Syhunt a partir de um script de pipeline do Jenkins, incluindo o Syhunt Dynamic, o Syhunt Code e o Syhunt Mobile no seu pipeline de entrega contínua, agendar varreduras e muito mais.

Onde instalar o Syhunt

O Syhunt pode ser instalado:

1. **Na própria máquina do Jenkins**, de modo que o Syhunt e o servidor Jenkins rodem lado a lado; ou
2. **Em uma máquina separada que atua como agente do Jenkins**, chamada remotamente via SSH pelo servidor Jenkins principal.

Usar agentes de CI é uma boa prática comum em ambientes DevOps, por isso recomendamos fortemente a segunda opção. Por segurança e desempenho, ambientes de produção também devem usar uma máquina dedicada como agente.

No Windows ou no Linux, **instale o Syhunt depois que o Jenkins estiver implantado e configurado**. Assim, o instalador do Syhunt detecta o Jenkins e instala automaticamente as extensões necessárias. Se você instalou o Syhunt primeiro, basta executar o instalador do Syhunt novamente.

Lembre-se de que o Syhunt é licenciado por dispositivo: cada máquina agente precisa de sua própria licença.

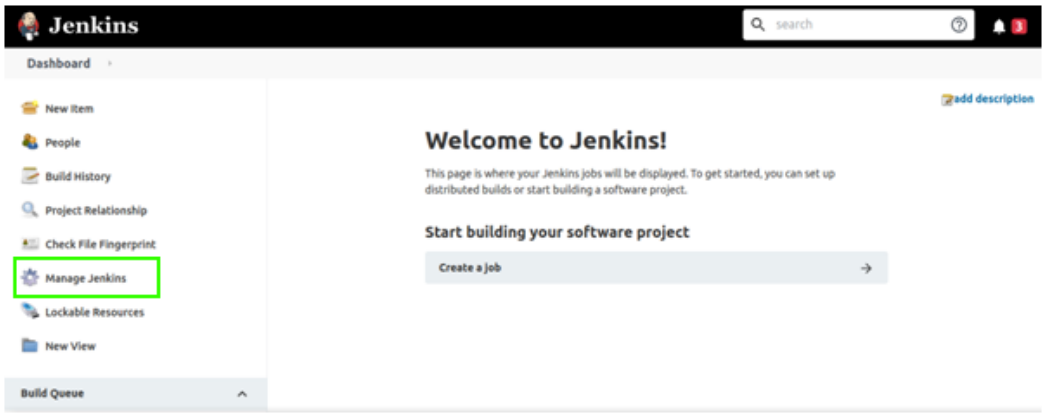
Ativando um agente do Syhunt

Com as instalações principal (controlador) e agente do Jenkins implantadas e configuradas, disponibilize o agente para o controlador:

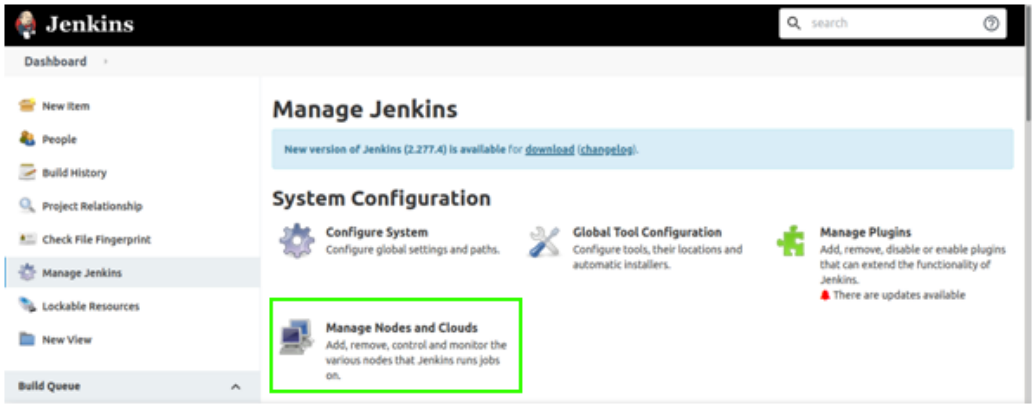
Página 122 de 182

Syhunt Hybrid 7.2

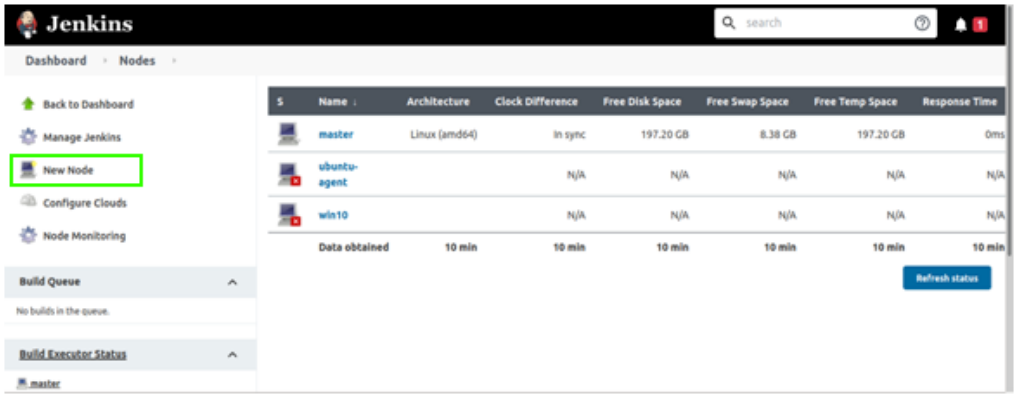
1. No painel do Jenkins do controlador, clique em **Manage Jenkins** na barra lateral.



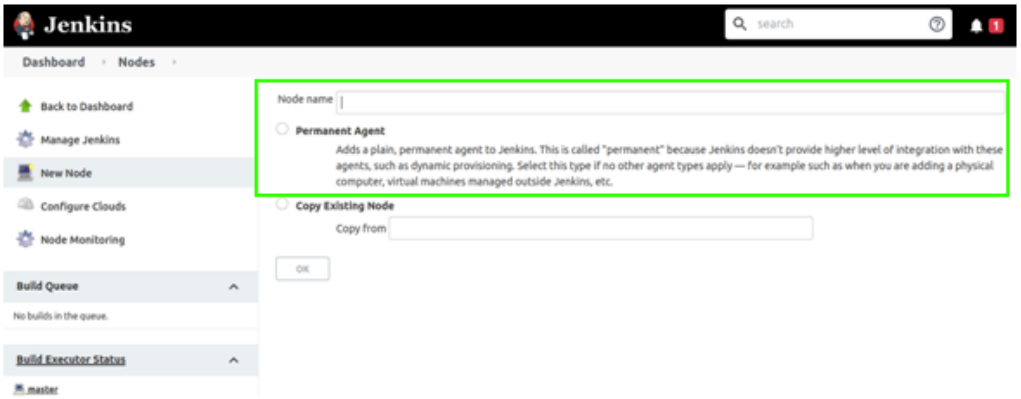
2. Em **System Configuration**, clique em **Manage Nodes and Clouds**.



3. Na tela **Nodes**, clique em **New Node** na barra lateral.



1. Dê um nome ao nó.



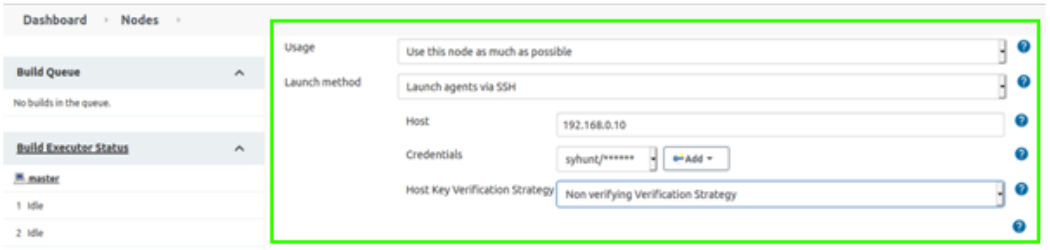
2. Selecione **Permanent Agent** e clique em **OK**.

Página 123 de 182

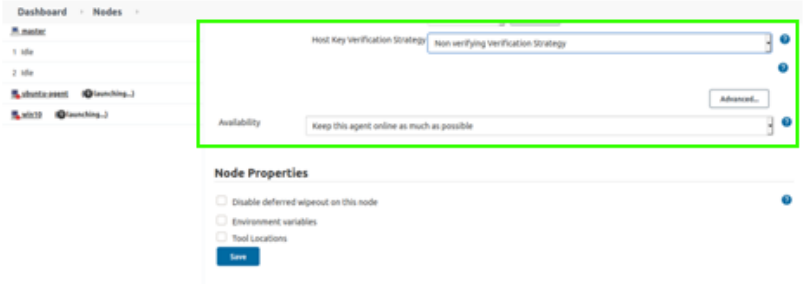
Syhunt Hybrid 7.2

3. Defina **# of executors** como **2**.

4. Defina **Remote root directory** como a pasta do Jenkins no agente: `/var/lib/jenkins` no Linux. No Windows, use a pasta `.jenkins`, normalmente `C:\Users\<seuusuário>\.jenkins` ou `C:\windows\system32\config\systemprofile\AppData\Local\Jenkins\.jenkins`; verifique qual delas existe no agente.
5. Dê ao nó um rótulo que será usado depois para referenciá-lo, por exemplo `Syhunt`.
6. Mantenha **Usage** no valor padrão.



7. Defina **Launch method** como **Launch agents via SSH**.
8. Informe o nome de host ou o endereço IP da máquina agente.
9. Em **Credentials**, clique em **Add** e informe o usuário e a senha do usuário Jenkins do agente.
10. Defina a **Host Key Verification Strategy**:
    - Se você não tiver uma credencial SSH, use **Non verifying Verification Strategy**.



    - Se você criou uma credencial SSH do Jenkins no agente, conforme a documentação do Jenkins, use **Manually provided key Verification Strategy**. Recomenda-se usar uma estratégia de verificação para mais segurança, principalmente em produção.
11. Mantenha **Availability** no valor padrão.
12. Clique em **Save**.

4. Se ainda não o fez, instale o Syhunt na máquina agente, conforme descrito abaixo.

Antes de instalar o Syhunt (Windows)

Depois de implantar e configurar o Jenkins:

1. Se for usar `scanCode()` para fazer a varredura de URLs Git, instale o Git for Windows.
2. Instale o Syhunt conforme descrito no guia de instalação.

Antes de instalar o Syhunt (Linux)

Depois de implantar e configurar o Jenkins, altere o usuário do Jenkins para o seu usuário atual (desde que não seja o `root`) ou para outro usuário de sua escolha. Não recomendamos executar o Jenkins como `root`, o que poderia expor o sistema operacional e os repositórios do Jenkins.

Página 124 de 182

1. Edite o arquivo de configuração do Jenkins:
   - CentOS: `sudo vi /etc/sysconfig/jenkins`
   - Ubuntu: `sudo vi /etc/default/jenkins`
2. Altere `JENKINS_USER="jenkins"` para `JENKINS_USER="usuariodesuaescolha"`.
3. Torne o novo usuário dono das principais pastas do Jenkins:

```
sudo chown -R usuariodesuaescolha:usuariodesuaescolha /var/lib/jenkins
sudo chown -R usuariodesuaescolha:usuariodesuaescolha /var/log/jenkins
sudo chown -R usuariodesuaescolha:usuariodesuaescolha /var/cache/jenkins
```

4. Reinicie o Jenkins:

```
sudo systemctl restart jenkins
```

5. Conectado com esse usuário, instale o Syhunt conforme descrito no guia de instalação para Linux. Se for usar `scanCode()` para fazer a varredura de URLs Git, instale também o Git (também explicado ali).

Adicionando o Syhunt a um pipeline

1. No painel do Jenkins, abra o pipeline.
2. Clique em **Configure** na barra lateral.
3. Na aba **Pipeline**, desmarque **Use Groovy Sandbox**, se estiver marcado.
4. Adicione um código como este no ponto adequado do seu script de pipeline:

```
def rootDir = pwd()
def syhunt = load "${rootDir}/../workspace@script/syhunt/syhunt.groovy"
syhunt.scanURL([target: 'http://somewebsite.com/', huntMethod: 'appscan', pfcond: 'fail-
if:risk=mediumup'])
```

   Um exemplo completo:

```
#!/usr/bin/env groovy

// Se não estiver usando um agente, altere a próxima linha para apenas: node {
node ("Syhunt") {
    stage('Scan') {
        def rootDir = pwd()
        def syhunt = load "${rootDir}/../workspace@script/syhunt/syhunt.groovy"
        syhunt.scanURL([target: 'http://somewebsite.com/', huntMethod: 'appscan', pfcond: 'fail-
if:risk=mediumup'])
    }
}
```

5. Clique em **Save**.

Executando o build

Quando o pipeline for executado, a saída do console será parecida com esta:

```
[Pipeline] echo
Preparing to scan URL: http://somewebsite.com/
[Pipeline] echo
VULNERABLE!
Found 2 vulnerabilities
SYHUNT URLSCAN 6.8.5.4 PLATINUM EDITION (c) 2020 Syhunt
...
[Pipeline] echo
Build problem: found Medium risk vulnerabilities.
[Pipeline] }
[Pipeline] // stage
```

Syhunt Hybrid 7.2

```
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: FAILURE
```

Função `scanURL`

Inicia uma varredura do Syhunt Dynamic.

Parâmetro	Descrição
`target` **(obrigatório)**	A URL a ser verificada, por exemplo `http://www.somesite.com`.
`huntMethod`	O método de varredura a ser usado. Se omitido, é usado o método padrão.
`pfcond`	Reprova o build quando uma condição de aprovação/reprovação é atendida.
`timelimit`	Tempo máximo da varredura, por exemplo `1d`, `3h`, `2h30m`, `50m`. A varredura é interrompida ao atingir o limite. Sem limite por padrão.

Retorna um mapa com:

Chave	Descrição
`outFilename`	O nome do arquivo do relatório gerado.
`outResultHTML`	O conteúdo do relatório gerado, como texto HTML.

Função `scanCode`

Inicia uma varredura do Syhunt Code.

Parâmetro	Descrição
`target` **(obrigatório)**	A URL de um repositório, ou uma pasta ou arquivo de código-fonte local.
`huntMethod`	O método de varredura a ser usado. Se omitido, é usado o método padrão.
`pfcond`	Reprova o build quando uma condição de aprovação/reprovação é atendida.
`branch`	O branch Git a ser verificado. Se omitido, o cliente Git obtém o branch padrão. Desnecessário para pastas ou arquivos locais.
`timelimit`	Tempo máximo da varredura, por exemplo `1d`, `3h`, `2h30m`, `50m`. Sem limite por padrão.

Retorna o mesmo mapa que `scanURL()`.

```
syhunt.scanCode([target: 'https://github.com/someuser/somerepo.git', huntMethod: 'normal', pfcond: 'fail-if:risk=mediumup'])
syhunt.scanCode([target: 'C:\\www\\', huntMethod: 'normal', pfcond: 'fail-if:risk=mediumup'])
```

Condições de aprovação/reprovação

Condição	Reprova se...
`high` ou `fail-if:risk=high`	for encontrada uma vulnerabilidade ou ameaça de risco Alto.
`medium` ou `fail-if:risk=mediumup`	for encontrada uma vulnerabilidade ou ameaça de risco Médio ou Alto.

Página 126 de 182

Syhunt Hybrid 7.2

Condição	Reprova se...
`low` ou `fail-if:risk=lowup`	for encontrada uma vulnerabilidade ou ameaça de risco Baixo, Médio ou Alto.

Agendamento de varreduras

1. No painel do Jenkins, abra o pipeline.
2. Clique em **Configure** na barra lateral.
3. Na aba **Build Triggers**, selecione **Build periodically** e informe o agendamento.
4. Clique em **Save**.

Página 127 de 182

Syhunt Hybrid 7.2

Rastreadores de Issues

> **Aplica-se a:** Syhunt Hybrid 7.2

Conteúdo

- GitHub Issues
- GitLab Issues
- Jira
- Enviando vulnerabilidades a um rastreador
- Nomes abreviados das preferências de rastreadores

Com um rastreador de issues configurado no Syhunt, você pode enviar vulnerabilidades a um projeto com um clique. Os rastreadores são criados como *conectores* na interface web; também é possível gerenciá-los pela linha de comando.

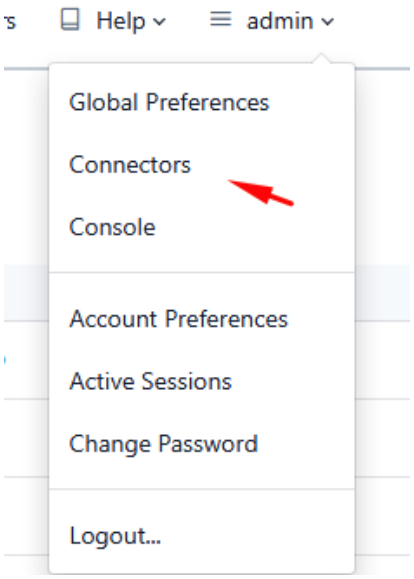
GitHub Issues

1. Crie um token de acesso pessoal com o escopo `repo`, se ainda não tiver um:

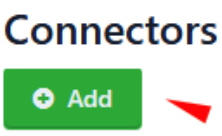
1. Faça login no GitHub.
2. Clique no seu avatar (canto superior direito) e escolha **Settings**.
3. No menu à esquerda, role para baixo e escolha **Developer settings**.
4. Escolha **Personal access tokens** e depois **Generate new token**.
5. Opcionalmente, informe uma descrição para o token.
6. Selecione o escopo `repo`.
7. Clique em **Generate token**.
8. Guarde o token em um local seguro. Depois de sair ou atualizar a página, não será possível vê-lo novamente.

2. Adicione um rastreador do GitHub:

1. Abra **Connectors** na interface web.



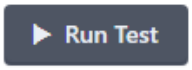
2. Clique em **Add**.



Página 128 de 182

Syhunt Hybrid 7.2

3. Informe um nome para o rastreador (por exemplo, `MyProjectName`).
4. Defina o tipo como **GitHub**.
5. Clique em **Create** e depois em **Close**.
6. Clique no novo rastreador na lista. A tela de preferências dele é aberta.
7. Informe o projeto do GitHub no formato `owner/repo`.
8. Informe seu token de acesso pessoal e clique em **Save**.
9. Abra o rastreador novamente e clique em **Run Test**.



Se tudo estiver configurado corretamente, uma issue de teste é criada em `https://github.com/<owner>/<repo>/issues`. Caso contrário, uma mensagem de erro indica o que corrigir.

GitLab Issues

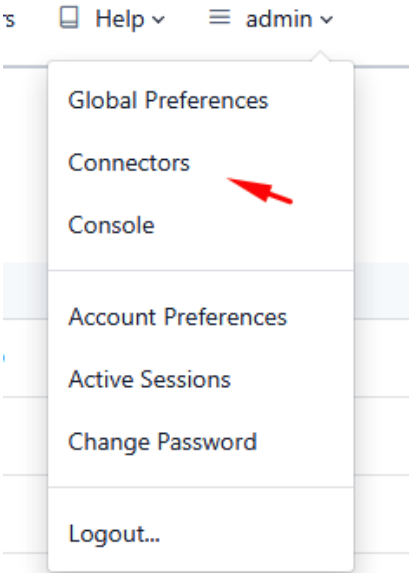
> **Dica**
> Para integrar o Syhunt ao GitLab CI e ao painel de segurança do GitLab, consulte GitLab.

1. Crie um token de acesso pessoal com o escopo `api`, se ainda não tiver um:

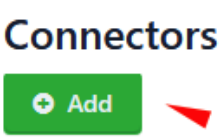
1. Faça login no GitLab.
2. Clique no seu avatar (canto superior direito) e escolha **Settings**.
3. No menu User Settings, escolha **Access Tokens**.
4. Informe um nome e, opcionalmente, uma data de expiração.
5. Selecione o escopo `api`.
6. Clique em **Create personal access token**.
7. Guarde o token em um local seguro. Depois de sair ou atualizar a página, não será possível vê-lo novamente.

2. Adicione um rastreador do GitLab:

1. Abra **Connectors** na interface web.

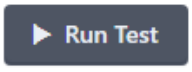


2. Clique em **Add**.



Página 129 de 182

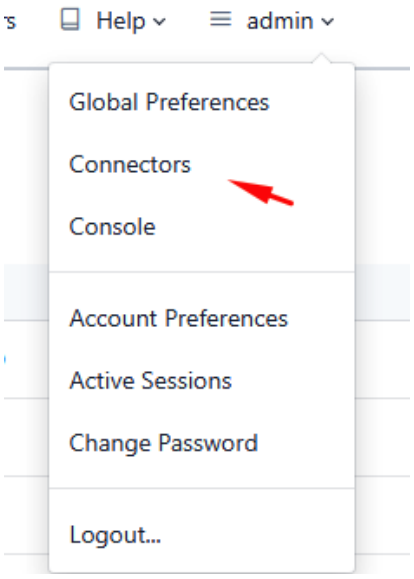
3. Informe um nome para o rastreador (por exemplo, `MyProjectName`).
4. Defina o tipo como **GitLab**.
5. Clique em **Create** e depois em **Close**.
6. Clique no novo rastreador na lista. A tela de preferências dele é aberta.
7. Informe o projeto do GitLab no formato `owner/repo`.
8. Informe a URL do servidor GitLab, por exemplo `https://gitlab.com/` ou a URL do seu próprio servidor.
9. Informe seu token de acesso pessoal e clique em **Save**.
10. Abra o rastreador novamente e clique em **Run Test**.



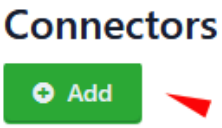
Se tudo estiver configurado corretamente, uma issue de teste é criada em `https://<servidor_gitlab>/<owner>/<repo>/issues`. Caso contrário, uma mensagem de erro indica o que corrigir.

Jira

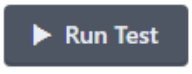
1. Abra **Connectors** na interface web.



2. Clique em **Add**.



3. Informe um nome para o rastreador (por exemplo, `MyProjectName`).
4. Defina o tipo como **JIRA**.
5. Clique em **Create** e depois em **Close**.
6. Clique no novo rastreador na lista. A tela de preferências dele é aberta.
7. Informe um nome de projeto Jira válido.
8. Informe a URL do servidor Jira, por exemplo `http://<IP>:8080/`.
9. Se não quiser o tipo de issue padrão, informe um nome de tipo de issue válido.
10. Informe suas credenciais do Jira (usuário e senha) e clique em **Save**.
11. Abra o rastreador novamente e clique em **Run Test**.

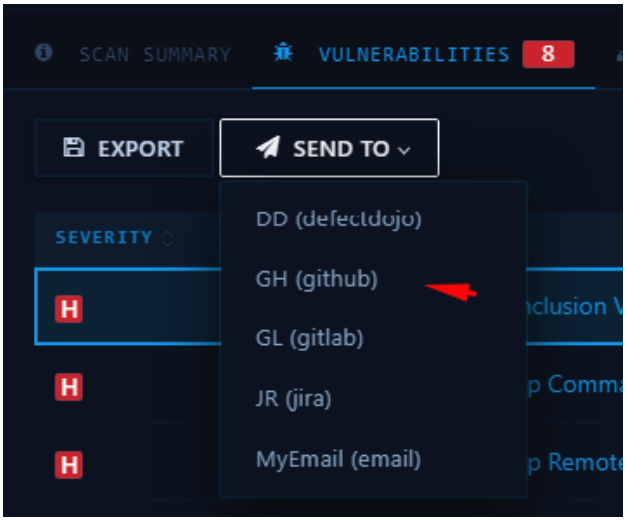


Se tudo estiver configurado corretamente, uma issue de teste é criada no seu projeto Jira. Caso contrário, uma mensagem de erro indica o que corrigir.

Syhunt Hybrid 7.2

Enviando vulnerabilidades a um rastreador

1. Abra a aba **Vulnerabilities** de uma varredura em andamento, ou abra **Scans**, clique em uma varredura na lista e vá para a aba **Vulnerabilities** dela.
2. Selecione uma ou mais vulnerabilidades, clique em **Send To** e escolha o rastreador.



O Syhunt também pode enviar automaticamente a um rastreador um resumo das vulnerabilidades encontradas:

- a partir de uma **varredura agendada**, se configurado na tela de agendamento;
- pela **CLI**, com o parâmetro `-tk` do `scanurl` ou do `scancode` (consulte Linha de Comando);
- pelo **PowerShell** ou por um **workflow de CI**, com o parâmetro `-tracker`:

```
# PowerShell, rastreador existente
Start-CodeScan -pfcond "medium" -tracker "YOURTRACKERNAME"

# PowerShell, rastreador definido na hora
Start-CodeScan -pfcond "medium" -tracker "?app=github###project=user/project###token=YOURTOKEN"

# Workflow do GitHub Actions, rastreador definido na hora
Start-CodeScan -pfcond "medium" -tracker "?app=github###project=${{ github.repository }}###token=${{
secrets.GITHUB_TOKEN }}"
```

Nomes abreviados das preferências de rastreadores

Ao informar preferências de rastreadores por meio de `-tk` ou `-tracker`, use os nomes abreviados abaixo. Configurações sem nome abreviado não podem ser definidas por esses parâmetros.

Nome abreviado	ID da preferência	Tipo
`app`	`tracker.defaultapp`	string
`project`	`tracker.project.name`	string
`token`	`tracker.auth.token.encrypted`	string
`from`	`tracker.message.from`	string
`to`	`tracker.message.tolist`	string
`subject`	`tracker.message.subject`	string
`notifyonfailonly`	`tracker.notifyonfailonly`	boolean
`apiurl`	`tracker.api.url`	string

Página 131 de 182

Syhunt Hybrid 7.2

Específicos do DefectDojo:

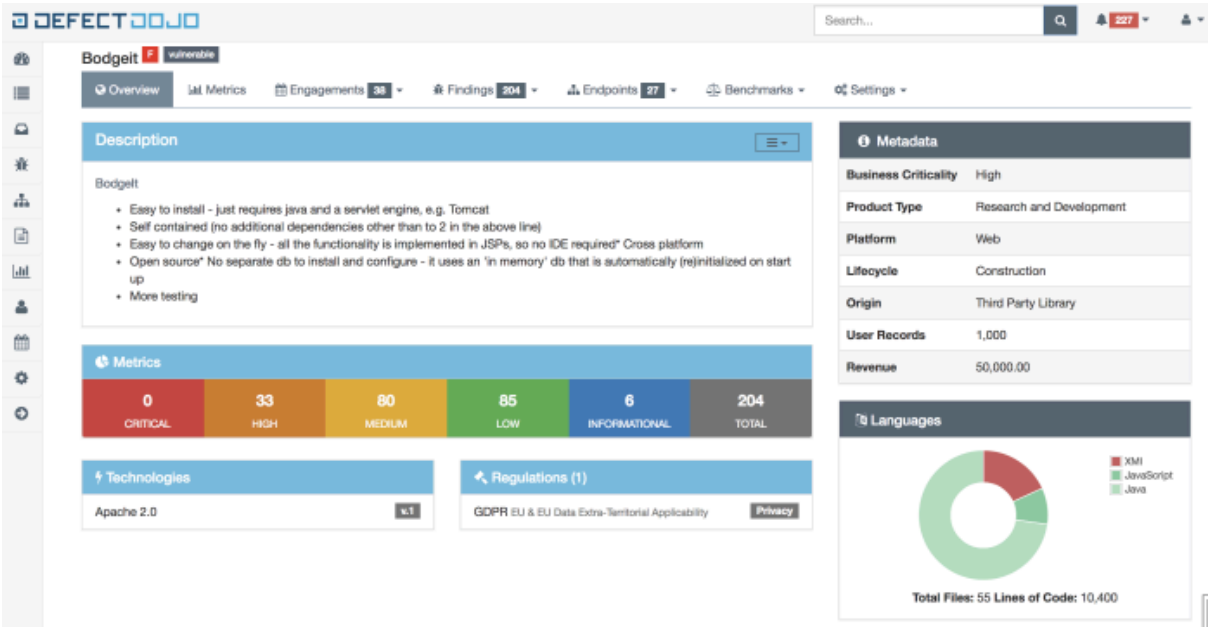
Nome abreviado	ID da preferência	Tipo
`producttypename`	`tracker.scan.producttypename`	string
`testtitle`	`tracker.scan.testtitle`	string
`engagementname`	`tracker.scan.engagementname`	string
`productname`	`tracker.scan.productname`	string
`minseverity`	`tracker.scan.minseverity`	string
`pushtojira`	`tracker.scan.pushtojira`	boolean
`verified`	`tracker.scan.verified`	boolean
`closeoldfindings`	`tracker.scan.closeoldfindings`	boolean
`skipduplicates`	`tracker.scan.skipduplicates`	boolean
`active`	`tracker.scan.active`	boolean
`autocreatecontext`	`tracker.scan.autocreatecontext`	boolean

Página 132 de 182

Syhunt Hybrid 7.2

DefectDojo

> **Aplica-se a:** Syhunt Hybrid 7.2



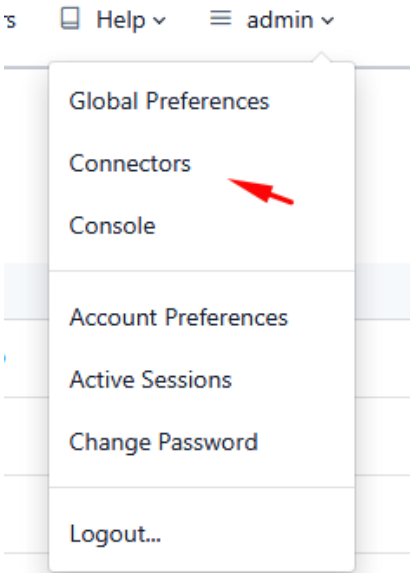
O DefectDojo automatiza a gestão de vulnerabilidades de segurança de aplicações. Ele agiliza os testes de segurança com recursos como importação de resultados de terceiros, mesclagem e remoção de duplicatas, integração com o Jira, modelos, geração de relatórios e métricas de segurança.

Desde a versão 6.9.17, o Syhunt pode se conectar à API de um servidor DefectDojo e enviar automaticamente resultados de SAST ou DAST. Como alternativa, o Syhunt pode gerar um relatório compatível para importação manual no DefectDojo.

Envio automático de resultados

1. Crie um conector do DefectDojo

1. Clique em **Connectors** no menu superior da interface web.



2. Clique em **Add**. A tela **New Connector** é aberta.

Página 133 de 182

3. Informe um nome para o conector (por exemplo, `MyDefectDojo`).
4. Defina **Type** como **DefectDojo**.
5. Clique em **Create** e, após a mensagem de sucesso, em **Close**.

2. Configure-o

1. Clique no conector na lista. A tela de preferências dele é aberta.
2. Informe a URL do servidor DefectDojo, por exemplo `https://<IP>/`.
3. Informe seu token de API.
4. Clique em **Save**.

3. Teste-o

Clique no conector na lista e escolha **Run Test**. Se tudo estiver configurado corretamente, uma varredura de teste aparece em `https://<servidor>/engagement/active`. Caso contrário, uma mensagem de erro indica o que corrigir.

4. Associe-o às suas varreduras

- **CLI:** informe `-tk:<nomedoconector>`.
- **Varreduras agendadas:** nas preferências da varredura agendada, na aba **Notifications**, selecione o conector em **Issue Tracker #1** ou **#2**.

Criando o conector pela CLI

```
-- Adicionar um rastreador do DefectDojo
scancore -tracker:add
-- Informe o tipo (DefectDojo) e um nome para o rastreador e pressione Enter.

-- Configurá-lo
scancore -tracker:set to:mytrackername -key:api.url -v:"https://serverurl/"
scancore -tracker:set to:mytrackername -key:auth.token.encrypted -vsecret

-- Testá-lo
scancore -tracker:send -tid:TEST -to:mytrackername

-- Configurações opcionais
scancore -tracker:set to:mytrackername -key:scan.minseverity -v:Info
scancore -tracker:set to:mytrackername -key:scan.testtitle -v:"Syhunt Scan"
scancore -tracker:set to:mytrackername -key:scan.engagementname -v:"Syhunt Hybrid Scan"
scancore -tracker:set to:mytrackername -key:scan.producttypename -v:"Research and Development"
scancore -tracker:set to:mytrackername -key:scan.productname -v:Syhunt
scancore -tracker:set to:mytrackername -key:scan.autocreatecontext -v:true
scancore -tracker:set to:mytrackername -key:scan.active -v:true
scancore -tracker:set to:mytrackername -key:scan.verified -v:true
scancore -tracker:set to:mytrackername -key:scan.closeoldfindings -v:false
scancore -tracker:set to:mytrackername -key:scan.skipduplicates -v:false
scancore -tracker:set to:mytrackername -key:scan.pushtojira -v:false
```

Em seguida, informe o rastreador ao PowerShell ou à CLI:

```
# PowerShell: rastreador existente com as configurações salvas
Start-CodeScan -pfcond "medium" -tracker "mytrackername"

# PowerShell: rastreador existente, alterando configurações apenas nesta execução
Start-CodeScan -pfcond "medium" -tracker "mytrackername?productname=Syhunt###autocreatecontext=true"

# PowerShell: rastreador definido na hora
Start-CodeScan -pfcond "medium" -tracker "?
app=defectdojo###apiurl=https://serverurl/###token=YOURTOKEN###productname=Syhunt###testtitle=Syhunt
Scan###autocreatecontext=true"
```

Pela CLI, use `-tk` em vez de `-tracker`:

```
scancode <alvo> -pfcond:medium -tk:"mydefectdojotrackername?productname=Syhunt###autocreatecontext=true"
```

Syhunt Hybrid 7.2

> **Observação**
> Em `-tk` e `-tracker`, as preferências devem usar seus nomes abreviados.

Gerando um relatório compatível

Para importar os resultados manualmente, faça o Syhunt gerar um relatório compatível com o DefectDojo dando ao arquivo de exportação a extensão dupla `.sast.json` ou `.dast.json`. Ao importar o arquivo no DefectDojo, escolha o formato **GitLab SAST** ou **GitLab DAST**.

Varredura	Exemplo
SAST	`scancode <alvo> -xout:qualquernome.sast.json`
DAST	`scanurl <alvo> -xout:qualquernome.dast.json`

- **GitHub, GitLab e PowerShell:** informe o nome do arquivo no parâmetro `outputex` da função de varredura.
- **Jenkins:** informe o nome do arquivo no parâmetro `outFilename` da função de varredura.

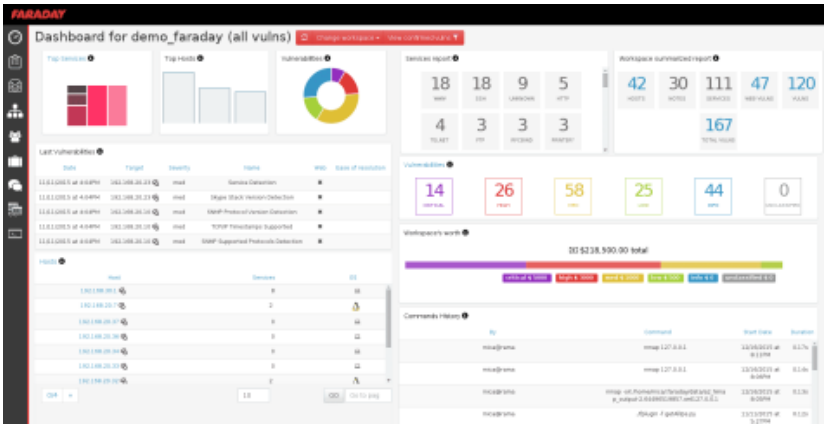
> **Observação**
> Atualmente, o DefectDojo documenta apenas o suporte ao GitLab SAST. A importação de DAST pode ter algumas limitações.

Página 135 de 182

Syhunt Hybrid 7.2

Faraday

> **Aplica-se a:** Syhunt Hybrid 7.2



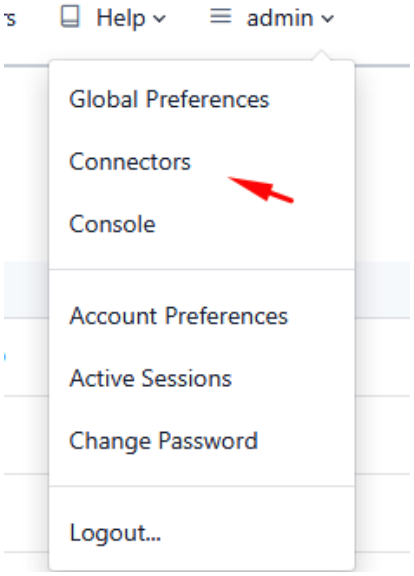
O Faraday permite que as equipes gerenciem sua superfície de ataque em um único lugar, automatizando e acelerando etapas importantes da gestão de vulnerabilidades. Ele reúne uma quantidade de ferramentas e procedimentos de segurança que, de outra forma, seria difícil de administrar, em um processo único e simples.

O Syhunt pode se conectar à API de um servidor Faraday e enviar automaticamente resultados de SAST ou DAST. O Faraday também inclui um plugin do Syhunt, de modo que os resultados do Syhunt podem ser enviados pela interface web, pela CLI e pela API REST do Faraday.

Envio automático de resultados

1. Crie um conector do Faraday

1. Clique em **Connectors** no menu superior da interface web.



2. Clique em **Add**. A tela **New Connector** é aberta.
3. Informe um nome para o conector (por exemplo, `MyFaraday`).
4. Defina **Type** como **Faraday**.
5. Clique em **Create** e, após a mensagem de sucesso, em **Close**.

2. Configure-o

Página 136 de 182

Syhunt Hybrid 7.2

1. Clique no conector na lista. A tela de preferências dele é aberta.
2. Informe a URL do servidor Faraday, por exemplo `https://yourdashboard.apps.faradaysec.com/`.
3. Informe o nome do seu workspace. Se deixar em branco, será usado `my_workspace`.
4. Informe seu usuário e senha.
5. Clique em **Save**.

3. Teste-o

Clique no conector na lista e escolha **Run Test**. Se tudo estiver configurado corretamente, uma varredura de teste aparece no seu painel do Faraday. Caso contrário, uma mensagem de erro indica o que corrigir.

4. Associe-o às suas varreduras

- **CLI:** informe `-tk:<nomedoconector>`.
- **Varreduras agendadas:** nas preferências da varredura agendada, na aba **Notifications**, selecione o conector em **Issue Tracker #1** ou **#2**.

Criando o conector pela CLI

```
-- Adicionar um rastreador do Faraday
scancore -tracker:add
-- Informe o tipo (Faraday) e um nome para o rastreador e pressione Enter.

-- Configurá-lo
scancore -tracker:set to:mytrackername -key:api.url -v:"https://yourdashboard.apps.faradaysec.com/"
scancore -tracker:set to:mytrackername -key:auth.username.encrypted -v:yourusername
scancore -tracker:set to:mytrackername -key:auth.password.encrypted -vsecret

-- Testá-lo
scancore -tracker:send -tid:TEST -to:mytrackername
```


Página 137 de 182

Syhunt Hybrid 7.2

Referência

Página 138 de 182

Syhunt Hybrid 7.2

Especificações Técnicas

> **Aplica-se a:** Syhunt Hybrid 7.2

Conteúdo

- A. Direitos de licença
- B. Arquitetura e base de vulnerabilidades
- C. Análise de código-fonte (SAST)
- C2. Análise mobile e de APK (MAST)
- D. Análise dinâmica (DAST)
- D2. Análise aumentada (OAST)
- D3. Análise híbrida aumentada (HAST)
- D4. Análise de APIs (AAST)
- D5. Análise forense de logs (FAST)
- E. Console de gerenciamento
- F. Recursos de varredura
- G. Requisitos de sistema
- H. Glossário

Este documento descreve as especificações técnicas do **Syhunt Hybrid**, uma ferramenta de teste de segurança de aplicações de alta precisão e ampla cobertura. Ela combina HAST aumentado (teste híbrido de segurança de aplicações), DAST (dinâmico), SAST (estático) e MAST (mobile) para identificar vulnerabilidades e fraquezas em aplicações.

A. Direitos de licença

- As licenças do Syhunt Hybrid normalmente têm validade de **12 meses**.
- As licenças incluem atualizações de versão e suporte técnico, além de documentação técnica em inglês.
- As licenças permitem um número ilimitado de varreduras de um número ilimitado de ativos por ano, iniciadas a partir de um único dispositivo (ou mais, se forem adquiridas estações adicionais). Cada licença inclui dois usuários da interface web, um administrador e mais um usuário; é possível adquirir usuários adicionais.
- Algumas licenças cobrem todas as linguagens de programação suportadas; outras cobrem pelo menos uma. É possível adquirir mais linguagens conforme a necessidade.
- As licenças permitem a integração com IA por meio de um endpoint de API configurável. Atualmente, o Syhunt suporta OpenAI, DeepSeek e modelos locais. Os custos e o hardware necessários para executar um modelo local ou na nuvem não são cobertos pelas licenças do Syhunt.

B. Arquitetura

O Syhunt Hybrid é uma ferramenta de avaliação de segurança de aplicações que é:

- **Híbrida:** combina várias técnicas de análise (DAST, OAST, SAST, MAST e FAST, descritas abaixo).
- **Local (on-premises):** é implantada nas instalações do cliente, de modo que o código-fonte analisado e os resultados nunca saem da rede interna.
- **Modular:** um único console de gerenciamento reúne a configuração e os resultados de todos os módulos.
- **Assíncrona:** executa várias varreduras ao mesmo tempo, sem esperar que cada uma termine.
- **Multiprocesso:** cada varredura roda como um processo separado do sistema operacional.
- **Acessível pela web e pela linha de comando:** tem uma interface web responsiva e uma CLI para iniciar varreduras dinâmicas e de código e outras tarefas.

Página 139 de 182

Syhunt Hybrid 7.2

Tipos de teste

Tipo	Descrição
SAST	Análise estática do código-fonte de aplicações web.
MAST	Análise estática do código-fonte de aplicações mobile (Android e iOS) e de arquivos APK do Android.
DAST	Análise dinâmica de aplicações web, com rastreamento profundo e injeção de parâmetros.
OAST	Análise dinâmica aumentada de aplicações web usando técnicas out-of-band (OOB).
HAST	Análise híbrida aumentada: os resultados da análise estática são usados automaticamente para aprimorar a análise dinâmica aumentada.
AAST	Análise dinâmica de APIs web, orientada por uma especificação de API.
FAST	Análise forense de arquivos de log de servidores web.

Base de vulnerabilidades

O Syhunt inclui uma base interna de vulnerabilidades que cobre as listas públicas de vulnerabilidades e os padrões abaixo. Ela é atualizada regularmente, para que a ferramenta acompanhe as novas vulnerabilidades publicadas por bases internacionais.

Padrão	Cobertura
CWE/SANS Top 25	Most Dangerous Software Errors, 2023 ou posterior
OWASP Top 10	Web Application Security Risks, 2021 ou posterior
OWASP Mobile Top 10	Mobile Risks, 2024 ou posterior (PLUS)
OWASP PHP Top 5	
CWE	Common Weakness Enumeration
CVE	Common Vulnerabilities and Exposures
ISO/IEC 27001	
WASC	Web Application Security Consortium Threat Classification
WAVSEP	Web Application Vulnerability Scanner Evaluation Project
NIST SAMATE	Projeto Software Assurance Metrics and Tool Evaluation
PCI DSS	Payment Card Industry Data Security Standard, 3.2, 3.2.1 ou posterior

Integrações

Categoria	Sistemas
Rastreamento de issues	Jira 7.13.0 ou posterior GitHub GitLab
Controle de versão	Repositórios Git públicos e privados Repositórios Azure DevOps e TFS (TFVC e Git) GitHub e GitLab Branches
APIs e scripts	API Lua 5.1 API Web (REST) PowerShell JSON e XML Selenium

Página 140 de 182

Syhunt Hybrid 7.2

Categoria	Sistemas
Painéis de segurança	OWASP DefectDojo GitLab Security Dashboard Faraday
Gestão de vulnerabilidades de aplicações (WAF)	F5 BIG-IP Application Security Manager (ASM) Imperva SecureSphere ModSecurity CRS OGASEC WAF
CI/CD	GitHub Actions GitLab CI e Security Dashboard Bitbucket Pipelines Jenkins Pipeline
Navegadores	Extensão Syhunt Sandcat (navegadores baseados no Chromium) Google Chrome Mozilla Firefox
E-mail	SMTP

Os exemplos abaixo mostram como fazer a varredura de um repositório Git por meio de cada integração:

```
-- Prompt de comando
scancode git://sub.domain.com/repo.git
scancode https://github.com/user/repo.git -rb:master
scancode https://dev.azure.com/user/projectname

-- Script do GitLab CI
- Start-CodeScan -pfcond 'fail-if:risk=mediumup' -output 'report.pdf' -outputex 'gl-sast-report.json'

-- Script do GitHub Actions
- Start-CodeScan -pfcond 'fail-if:risk=mediumup' -output 'report.pdf'

-- Script de pipeline do Jenkins
syhunt.scanCode([target: 'https://github.com/someuser/somerepo.git', branch: 'master', pfcond: 'fail-
if:risk=mediumup'])

-- PowerShell
$MyScan = @{
  target = 'https://github.com/someuser/somerepo.git';
  branch = 'master';
  pfcond = 'fail-if:risk=mediumup';
  output = 'report.pdf'
}
Start-CodeScan @MyScan

-- Script Lua
code:scanurl('https://github.com/someuser/somerepo.git', 'master')

-- API Web (REST, JSON), POST /syhunt/launch.lua
{
  "sourcetarget": "https://github.com/someuser/somerepo.git",
  "sourcebranch": "master",
  "apikey": "YOUR_API_KEY"
}
```

Linguagens e ambientes suportados

Linguagem	Ambientes e frameworks
C#	ASP.NET Web Forms (ASPX) ASP.NET Core MVC .NET MAUI (iOS e Android) Xamarin (iOS e Android) Blazor Razor Pages

Página 141 de 182

Syhunt Hybrid 7.2

Linguagem	Ambientes e frameworks
Dart	Flutter (iOS e Android) Lado do servidor
Java	JEE JSP Android Spring Framework
JavaScript	Lado do cliente Lado do servidor Node.js (puro, Express.js e Koa.js) Angular (2 ou posterior) AngularJS JScript (ASP Classic) Electron (desktop)
Kotlin	Ktor
Lua	ngx_lua mod_lua CGILua Lua Pages
Objective-C, C e C++	iOS
Object Pascal	Delphi Delphi XE Lazarus DWS
Perl	
PHP	
Python	CGI mod_python PSP WSGI Django
Ruby	Rails ERB mod_ruby
Swift	iOS
TypeScript	
VB	VB.NET (ASP.NET) VBScript (ASP Classic)

C. Análise de código-fonte (SAST)

Módulos	Syhunt Code; Syhunt Mobile
Descrição	Analisa o código-fonte de aplicações web e mobile em busca de vulnerabilidades de segurança.
Tipos de análise	SAST (Static Application Security Testing) MAST (Mobile Application Security Testing)
Aplicações alvo	Aplicações web (incluindo web services) Aplicações mobile (Android e iOS) Aplicações desktop

Página 142 de 182

Syhunt Hybrid 7.2

Linguagens e plataformas alvo	APK (Android) ASP.NET e ASP Classic .NET MAUI / Xamarin (iOS e Android) Blazor / Razor Pages Dart (Android e iOS) HTML Java (incluindo Android) JavaScript Kotlin (incluindo Android) Lua Objective-C, C e C++ (iOS) Object Pascal (Delphi) Perl PHP Python Ruby Swift (iOS) TypeScript
Integração com Git	Sim

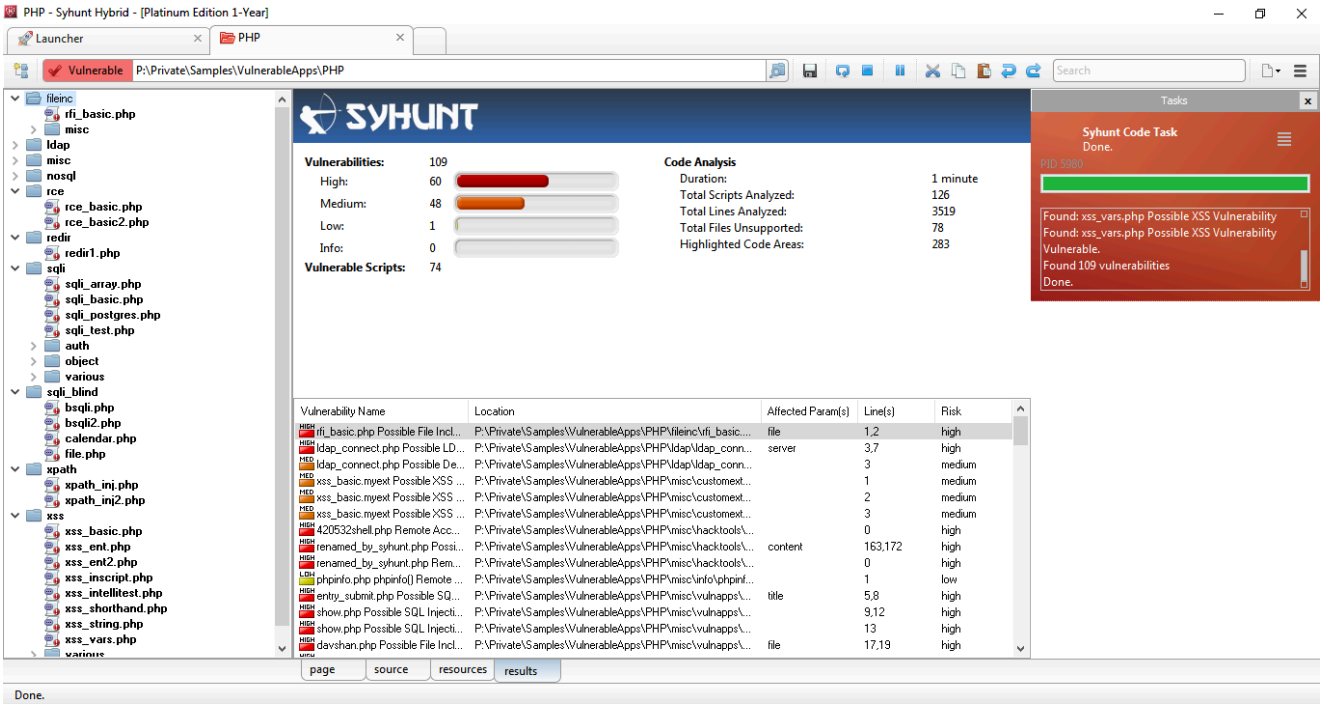
Recursos:

- Detecta vulnerabilidades conhecidas de aplicações em todas as linguagens e ambientes suportados.
- Suporta código embutido em HTML, atalhos de impressão (print shorthands) e interpolação de strings.
- Encontra vulnerabilidades tanto do lado do cliente quanto do lado do servidor.
- Detecta scripts desatualizados e vulneráveis, locais ou remotos, como versões vulneráveis de AngularJS, jQuery, fullPage, Bootstrap e Moment.js.
- Analisa bases de código completas, trechos de código e arquivos de configuração.
- Realiza análise incremental: os principais resultados e dados das varreduras de um código são armazenados automaticamente e usados para acelerar as varreduras seguintes.
- Audita várias linguagens de programação na mesma varredura.
- Suporta TypeScript, detectando vulnerabilidades antes que o código seja compilado para JavaScript (lado do cliente e do servidor, Node.js e Angular).
- Identifica áreas-chave do código, como marcadores HTML específicos, JavaScript, requisições XHR, pontos de entrada e palavras-chave relevantes, e permite navegar entre elas.
- Encontra vulnerabilidades causadas por código mal projetado, ou seja, erros de programação que expõem o sistema a ataques.
- Detecta **shadow AI**, incluindo o uso de endpoints de API da DeepSeek, da OpenAI, compatíveis com OpenAI e de outros sistemas de IA emergentes.
- Analisa arquivos de configuração para avaliar ameaças e identificar contramedidas no nível de configuração do servidor, do ambiente ou da aplicação.
- Reconhece filtragem e validação de entradas, para resultados precisos e livres de falsos positivos.
- *Com IA:* adiciona aos relatórios exemplos de código corrigido, ao lado de cada trecho de código vulnerável.
- *Com IA:* adiciona aos relatórios uma pontuação de confiança para cada resultado, a probabilidade estimada (em porcentagem) de que seja um verdadeiro positivo e não um falso positivo.

Página 143 de 182

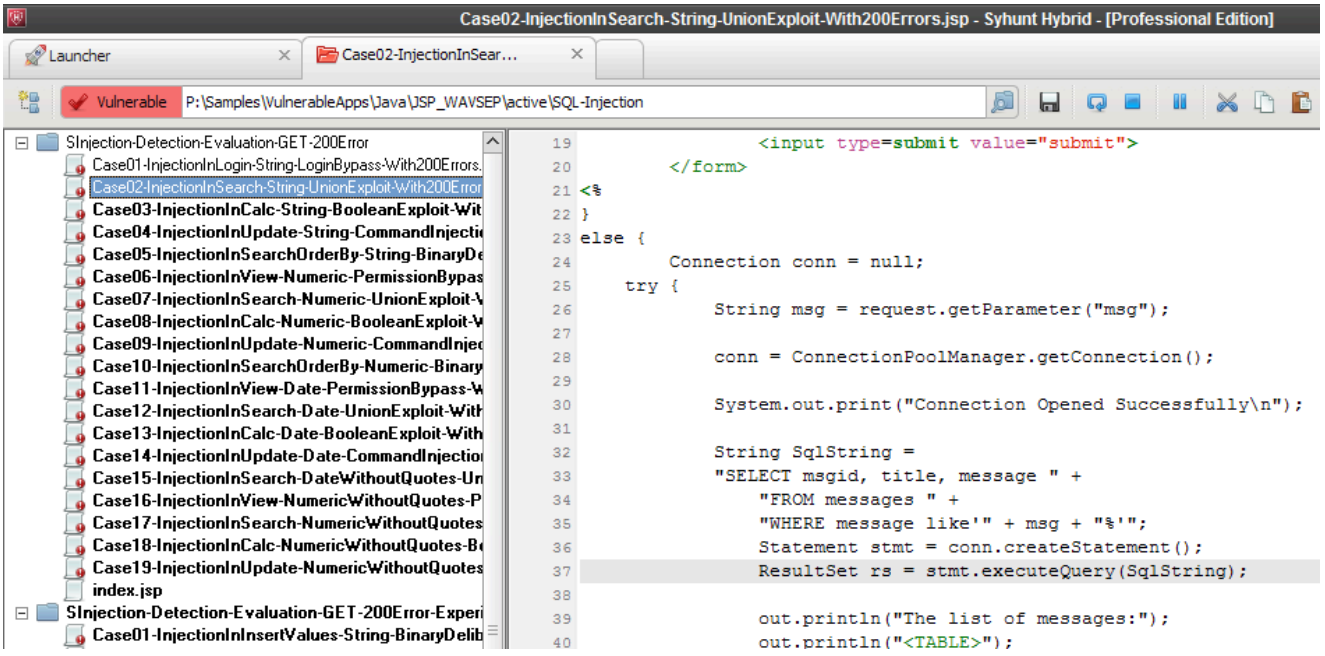
Syhunt Hybrid 7.2

Visualização da varredura em tempo real



A interface mostra em tempo real o status das varreduras de código em andamento, incluindo:

- a árvore de arquivos analisados e vulneráveis;
- a duração da varredura;
- o número de scripts analisados e vulneráveis;
- o número de vulnerabilidades encontradas, no total e por severidade;
- o número de linhas analisadas;
- a lista de vulnerabilidades encontradas.



Ela também destaca os pontos do código em que uma única correção resolveria várias das vulnerabilidades encontradas.

Página 144 de 182

Syhunt Hybrid 7.2

C2. Análise mobile e de APK (MAST)

O Syhunt Code realiza testes de segurança de aplicações mobile no código-fonte Android e iOS e em arquivos APK do Android, cobrindo os principais riscos do OWASP Mobile Top 10 e do CWE/SANS Top 25, como:

1. Uso inadequado de credenciais
2. Segurança inadequada da cadeia de suprimentos
3. Autenticação/autorização inseguras
4. Validação insuficiente de entrada/saída
5. Comunicação insegura
6. Controles de privacidade inadequados
7. Proteções insuficientes do binário
8. Configuração de segurança incorreta
9. Armazenamento de dados inseguro
10. Criptografia insuficiente

D. Análise dinâmica (DAST)

Módulo	Syhunt Dynamic Augmented
Descrição	Detecta vulnerabilidades de segurança em aplicações web dinâmicas e em servidores web.
Tipos de análise	DAST (Dynamic Application Security Testing) OAST (Out-of-Band Application Security Testing)
Aplicações alvo	Aplicações web e servidores web
Linguagens e plataformas alvo	ASP.NET e ASP Classic HTML e JavaScript Java Lua Perl PHP Python Ruby
Otimizado para servidores HTTP	Apache Apache Tomcat Microsoft IIS NGINX OpenResty
Protocolos	HTTP 1.0 e 1.1 HTTPS (SSL 2, SSL 3, TLS 1 a TLS 1.3) IPv6 Proxies (HTTP, SOCKS 4 e SOCKS 5)
Recursos HTTP	Keep-Alive Compressão GZIP Autenticação: Basic, Bearer, Digest (MD5/SHA-256/SHA-512), por formulário e NTLM*

* O NTLM está disponível apenas no Windows. O suporte ao Kerberos está planejado.

Recursos:

- Detecta vulnerabilidades conhecidas de aplicações web em todas as linguagens e ambientes suportados.
- Realiza **análise dinâmica aumentada**, combinando DAST e OAST.
- Realiza **análise híbrida aumentada**, combinando DAST, OAST e SAST.
- Realiza análise incremental: os principais resultados e dados das varreduras de uma URL alvo são armazenados automaticamente e usados para acelerar as varreduras seguintes.

Página 145 de 182

Syhunt Hybrid 7.2

- Suporta uma ampla variedade de métodos de autenticação, incluindo detecção automática do formulário de login (Smart e Standard), logins gravados no navegador (Recorder), logins baseados em Selenium (Selenium-Auto e scripts Node.js), autenticação HTTP e login manual pela extensão Sandcat.
- Informa se uma sessão autenticada se manteve do início ao fim de uma varredura autenticada.
- Detecta scripts, software de servidor e outros componentes desatualizados e vulneráveis.
- Encontra vulnerabilidades tanto do lado do cliente quanto do lado do servidor.
- Encontra vulnerabilidades in-band, inferenciais e out-of-band (OOB).
- Detecta aplicações vulneráveis conhecidas em todas as linguagens suportadas, incluindo ColdFusion, Flash e Server-Side Includes (SSI).
- Detecta **shadow AI** por meio de técnicas de injeção, incluindo o uso da DeepSeek, da OpenAI e de outros sistemas de IA emergentes.
- Realiza automaticamente ataques de força bruta à estrutura, a formulários de login e à autenticação HTTP.
- Detecta automaticamente páginas de logout e as evita durante a varredura, com configuração manual opcional.
- Suporta varreduras invasivas e não invasivas.
- Permite desativar testes de negação de serviço (DoS) que possam afetar a disponibilidade da aplicação.
- Permite configurar o número de novas tentativas e o tempo limite de conexão com a aplicação.
- Suporta aplicações construídas sobre sistemas de gerenciamento de conteúdo como Drupal, Joomla, Wix e WordPress.
- Suporta aplicações de página única (SPAs).
- Identifica as tecnologias usadas pela aplicação e otimiza o tempo de varredura de acordo com elas.
- Identifica versões ocultas de software de servidor e de componentes (Hunter-Sense™), como Apache, NGINX, PHP, mod_ssl, OpenSSL e Phusion Passenger.
- Injeta dados e manipula parâmetros em URLs e formulários (GET e POST).
- Aplica mutações de injeção para cobrir todas as linguagens e plataformas suportadas.
- Realiza verificações de injeção baseadas em tempo com taxa de falsos positivos de 0% para injeção de código, injeção de expression language, injeção de NoSQL, execução remota de comandos e injeção de SQL.
- Encontra vulnerabilidades como injeção de SQL, injeção de NoSQL, injeção de comandos do sistema operacional, divulgação e injeção de código usando técnicas inferenciais (tempo de resposta), in-band (mensagem de erro ou impressão), out-of-band (OOB) e passivas.
- Inclui um rastreador AJAX opcional para rastreamento profundo de aplicações com uso intenso de AJAX, construídas com frameworks populares como Angular, AngularJS e React, com suporte a pré-renderização e a local storage.

Navegação e rastreamento

O Syhunt mapeia a estrutura da aplicação alvo, incluindo todos os links e pontos de entrada, emulando um navegador moderno com suporte a:

Área	Suporte
Padrões web	HTML5 CSS3
Análise de HTML	Caminhos relativos HTML padrão HTML fora do padrão ou malformado Reconhecimento de formulários
Emulação de navegador	Google Chrome Mozilla Firefox Microsoft Edge Envio de Referer Pré-renderização Local storage
JavaScript	Análise e execução de JavaScript Emulação de DOM Comportamento de navegador Chamadas XHR (XMLHttpRequest) / AJAX Arquivos JavaScript externos

Página 146 de 182

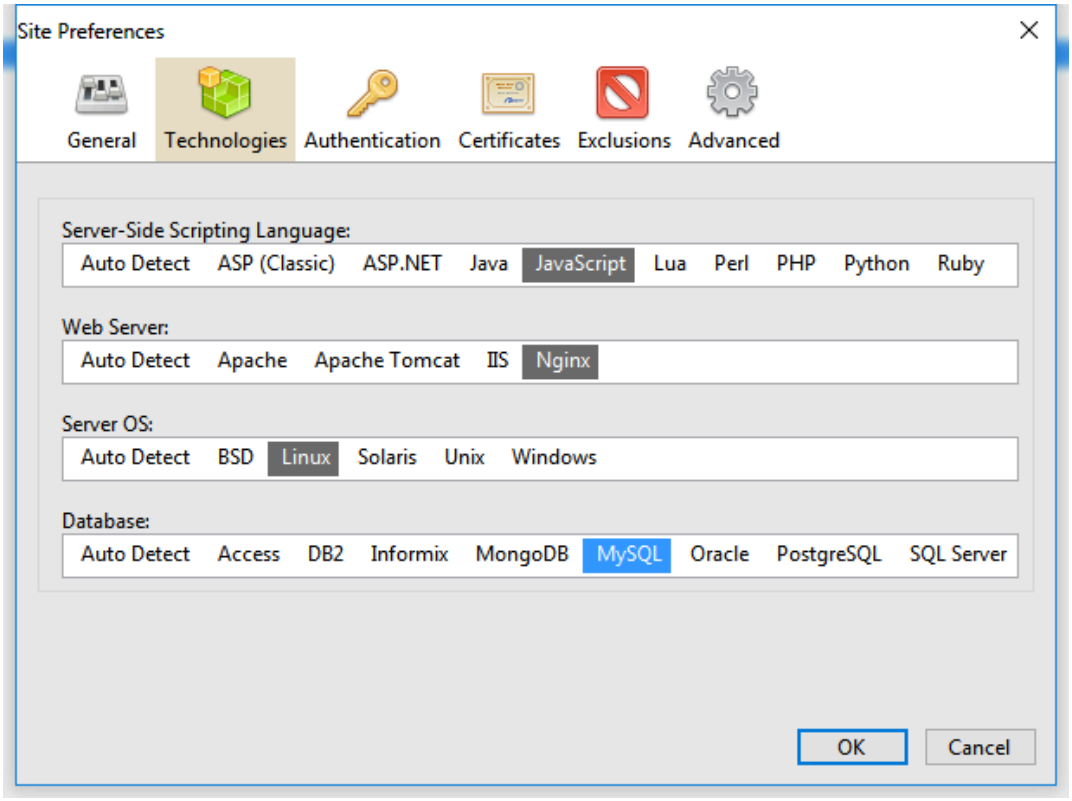
Syhunt Hybrid 7.2

Área	Suporte
Simulação de interação do usuário	Pressionamento de teclas Cliques do mouse Preenchimento automático de formulários Login automático Prevenção de logout acidental
Cookies	Gerenciamento de cookies Gerenciamento de sessões
Redirecionamentos	HTTP Meta refresh JavaScript
Isolamento de processos	Cada varredura roda como um processo separado
robots.txt	Analisado quando disponível

O Syhunt lida com sites grandes e complexos, com conteúdo gerado dinamicamente, e evita loops ao mapear a aplicação. Também é possível limitar a profundidade da varredura, incluindo:

- o número máximo de links por servidor e por página;
- o tamanho máximo de URL (bytes) e de resposta HTTP (kilobytes).

Preferências de site



O Syhunt Dynamic permite criar perfis de varredura para os ativos, especificando:

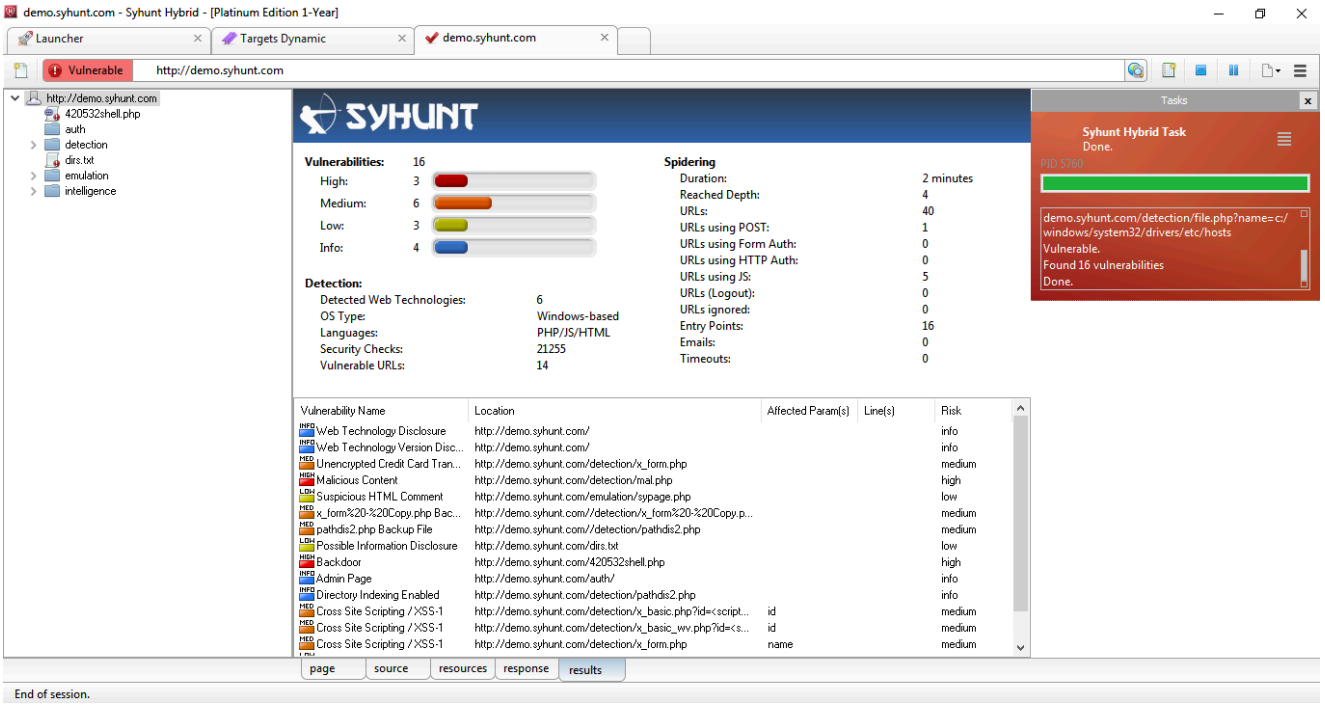
- as tecnologias usadas pela aplicação (linguagem do lado do servidor, servidor web, sistema operacional e banco de dados), para otimizar o tempo de varredura;
- URLs iniciais;
- certificados SSL e credenciais de autenticação, para fazer a varredura com login no alvo;
- objetos a excluir da varredura, como caminhos, formulários e vulnerabilidades específicos;
- limites de profundidade e de camadas;
- assinaturas para detectar páginas de erro 404 personalizadas;

Página 147 de 182

Syhunt Hybrid 7.2

- cookies e tokens de sessão configurados manualmente.

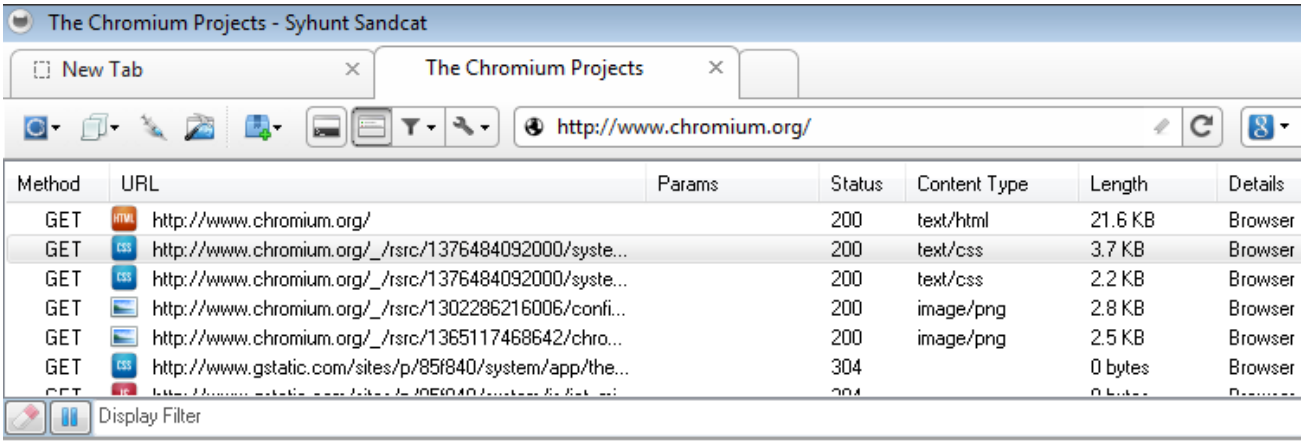
Visualização da varredura em tempo real



A interface mostra em tempo real o status das varreduras dinâmicas em andamento, incluindo:

- a árvore do site com os caminhos encontrados e vulneráveis (mapa e estrutura do site);
- a duração da varredura;
- o número de URLs vulneráveis;
- o número de URLs que usam POST, autenticação e JavaScript;
- o número de pontos de entrada;
- o número de tempos limite atingidos (timeouts);
- o número de vulnerabilidades encontradas, no total e por severidade;
- o número de verificações de segurança realizadas;
- as tecnologias web detectadas;
- o tipo de sistema operacional do alvo;
- as linguagens de programação do alvo;
- a profundidade alcançada;
- a lista de vulnerabilidades encontradas.

Testes manuais e assistidos pelo navegador



Página 148 de 182

Syhunt Hybrid 7.2

O Syhunt Dynamic suporta testes manuais e assistidos pelo navegador. Desde a versão 7.2, isso é feito pela extensão Syhunt Sandcat para navegador; as versões 7.1 e anteriores incluíam o navegador integrado Sandcat Browser. Os recursos incluem:

- login manual em aplicações web;
- varredura após login com CAPTCHAs, tokens e outros mecanismos de autenticação de dois fatores;
- varredura após rastreamento/navegação manual;
- captura de URLs e pontos de entrada.

O Sandcat Browser clássico (7.1 e anteriores) também oferecia cabeçalhos HTTP ao vivo; pré-visualização dos formatos de arquivo web mais comuns (CSS, Flash, HTML, BMP, GIF, ICO, JPG, PNG, SVG, JavaScript, JSON, texto e XML); desofuscação automática de JavaScript; repetição de requisições; um fuzzer; um executor de scripts; e editores HTTP/XHR com carregador de requisições.

D2. Análise aumentada (OAST)

Módulo	Syhunt Signal
Descrição	Detecta vulnerabilidades de segurança out-of-band em aplicações web dinâmicas.
Tipo de análise	OAST (Out-of-Band Application Security Testing)
Aplicações alvo	Aplicações web

O Syhunt Dynamic Augmented trabalha com o serviço online **Syhunt Signal** para realizar testes out-of-band automatizados. O Syhunt Signal:

- permite ao scanner detectar variantes de alto risco de vulnerabilidades out-of-band (OOB) que, de outra forma, seriam invisíveis;
- escuta as requisições forçadas a partir de um servidor alvo vulnerável durante a varredura e as sinaliza de volta ao scanner;
- correlaciona automaticamente os alertas recebidos com as requisições de ataque que os provocaram;
- adiciona as vulnerabilidades OOB encontradas ao relatório e à interface;
- não gera falsos positivos;
- extrai automaticamente dados de um alvo vulnerável, usando comandos e técnicas específicos do ambiente e do sistema operacional, e os adiciona aos resultados da varredura;
- exige uma conexão ativa com a Internet.

Tipos de vulnerabilidades out-of-band detectados pelo Syhunt Signal:

- Execução de comandos
- Inclusão de arquivo remoto (RFI)
- Server-side request forgery (SSRF)
- Injeção de SQL
- Injeção de entidade externa XML (XXE)

D3. Análise híbrida aumentada (HAST)

O Syhunt Hybrid combina os resultados do Syhunt Dynamic, do Syhunt Signal e do Syhunt Code para realizar a **análise híbrida aumentada** (HAST aumentado), na qual os resultados da análise estática são usados automaticamente para aprimorar a análise dinâmica aumentada. Ela:

- combina SAST, DAST e OAST;
- realiza análise híbrida do código JavaScript do lado do cliente (SAST-in-DAST);
- detecta e confirma vulnerabilidades dinamicamente, simulando ataques inferenciais, in-band e out-of-band e usando pontos de entrada e outras informações obtidas do código-fonte.

Página 149 de 182

Syhunt Hybrid 7.2

D4. Análise de APIs (AAST)

Módulo	Syhunt API
Descrição	Detecta vulnerabilidades de segurança em APIs web.
Tipos de análise	AAST (API Application Security Testing) OAST (Out-of-Band Application Security Testing)
Aplicações alvo	APIs web
Formatos de especificação	API Blueprint, OpenAPI 2 e 3, Swagger 1 a 3, Postman Collection 1 a 2.1, GraphQL (com anotações REST), I/O Docs, Google Discovery, RAML, WADL

Uma varredura de API começa a partir de uma especificação de API, em vez de uma URL inicial. O Syhunt mapeia a API a partir da especificação, valida se ela se comporta conforme especificado e então testa seus endpoints com as mesmas técnicas de injeção usadas nas varreduras dinâmicas, cobrindo tanto o OWASP Top 10 quanto o OWASP API Security Top 10. Consulte Varredura de APIs Web.

D5. Análise forense de logs (FAST)

Módulo	Syhunt Forensic
Descrição	Detecta ataques contra aplicações web dinâmicas.
Tipo de análise	FAST (Forensic Application Security Testing)
Aplicações alvo	Aplicações web

O Syhunt Forensic realiza, sob demanda, análises heurísticas de segurança de logs de servidores web para detectar ataques, incluindo:

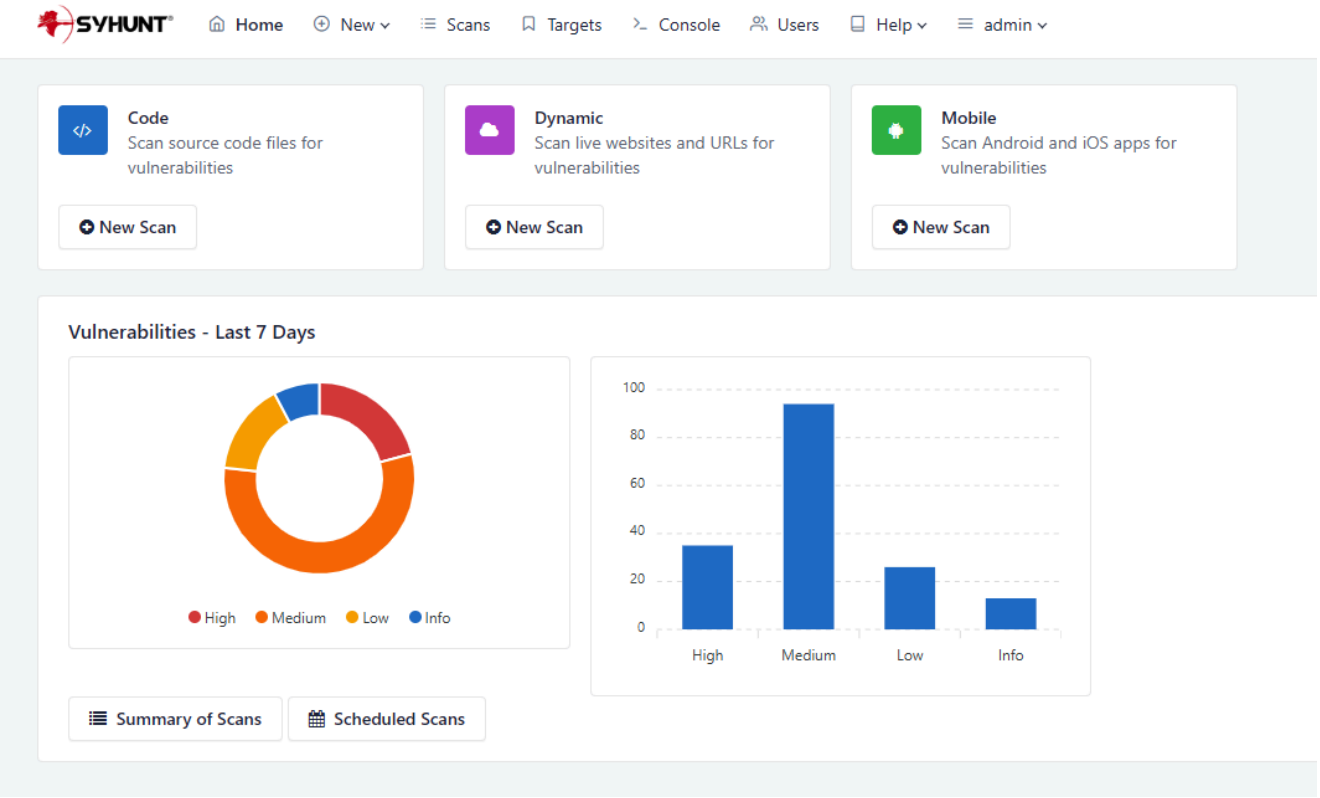
- identificação da origem (endereço IP), do país, do tipo e dos métodos das tentativas de comprometer aplicações web;
- reconstrução de sessões de ataque, separando com precisão o tráfego legítimo do malicioso e os ataques automatizados dos manuais;
- detecção de intrusões, de instalação de backdoors e das ferramentas de intrusão usadas;
- detecção de tentativas de explorar vulnerabilidades do OWASP Top 10 e do uso de técnicas de evasão de defesas;
- suporte a logs do Apache, do Microsoft IIS e do NGINX, com detecção automática de formato.

Consulte Análise Forense de Logs.

Página 150 de 182

Syhunt Hybrid 7.2

E. Console de gerenciamento



O console de gerenciamento reúne as configurações dos módulos e os resultados das varreduras SAST, DAST e FAST. Ele:

- dá acesso aos resultados armazenados de varreduras concluídas e em andamento;
- pode ser usado sem conhecimento prévio de segurança da informação ou de programação segura;
- exige pouca ou nenhuma intervenção do usuário antes e durante as varreduras;
- mostra graficamente o progresso e o nível de risco de cada análise em andamento;
- mostra as informações da licença: tipo de licença, data de expiração e linguagens de programação suportadas;
- permite desativar regras de detecção de vulnerabilidades e ver quais regras estão desativadas;
- permite ignorar uma ou mais vulnerabilidades específicas;
- permite pausar, retomar e cancelar varreduras imediatamente;
- permite localizar, excluir, exportar e importar resultados de varreduras;
- permite adicionar e gerenciar ativos;
- permite exportar e importar listas de alvos como arquivos CSV ou de lista;
- permite exportar e importar as configurações da ferramenta;
- mostra em tempo real a lista de vulnerabilidades encontradas;
- gera relatórios resumidos de varreduras por método de varredura;
- lista todas as sessões de um determinado mês;
- realça a sintaxe de todas as linguagens de programação suportadas;
- *(interface clássica)* inclui um bloco de notas para testes manuais, com geradores de strings de injeção comuns, geradores de hash, codificadores e decodificadores, funções HTML e manipulação de texto;
- inclui um Extension Development Kit (EDK) para adicionar novos recursos.

Controle de acesso de usuários

- A interface web aplica uma política de senhas rigorosa:
  - comprimentos mínimos para senhas de administradores e demais usuários;
  - proibição de reutilizar senhas recentes;
  - regras de complexidade: sem o nome de usuário ou parte do nome completo do usuário; pelo menos uma letra maiúscula, uma letra minúscula, um dígito e um caractere não alfanumérico; sem três ou mais caracteres idênticos seguidos (como `AAA`) e sem três caracteres consecutivos em ordem crescente ou decrescente (como `123`, `321`,

Página 151 de 182

Syhunt Hybrid 7.2

`ABC`, `CBA`).

- A interface web tem proteção integrada contra força bruta, gera IDs de sessão de forma segura e armazena as senhas com hash seguro.
- Acesso baseado em perfis: Administrator, Operator, Restricted Operator e Viewer.

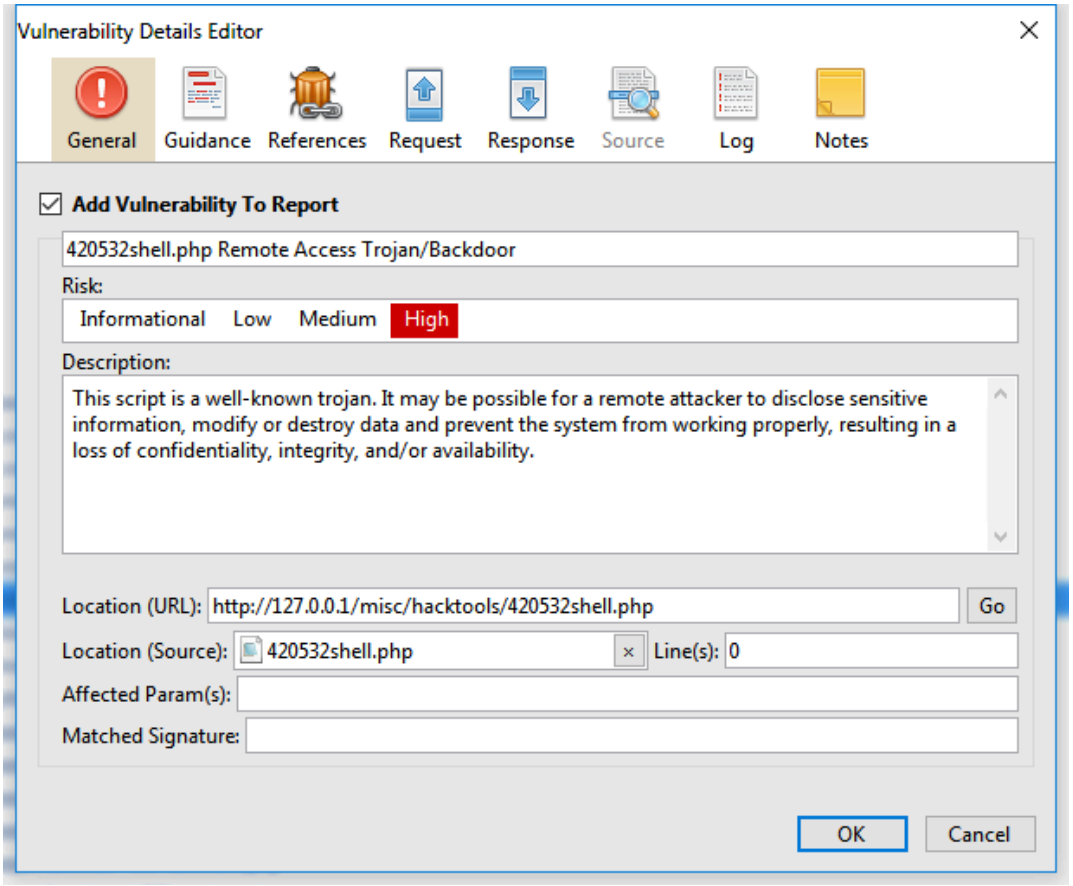
Agendamento de varreduras

As varreduras agendadas suportam:

- horário, data ou dia da semana;
- tipo de varredura (dinâmica ou de código);
- alvo da varredura;
- tempo limite da varredura;
- geração de relatório, com escolha do modelo;
- envio do relatório por e-mail ao final da varredura;
- notificações para rastreadores de issues ou painéis de segurança;
- uma condição de reprovação baseada na severidade mínima;
- varreduras ocultas (sem janela);
- exportação da linha de comando para uso em agendadores de terceiros e outras ferramentas.

Alertas de vulnerabilidades

O console de gerenciamento permite visualizar e editar alertas de vulnerabilidades, incluindo alterar seu status e severidade e adicionar comentários.



Cada alerta inclui:

- nome e descrição da vulnerabilidade;
- localização (uma URL ou um arquivo);
- referências a bases de vulnerabilidades conhecidas, como CVE, CWE, NVD e OSVDB, se houver;
- pontuação CVSS (versões 2 e 3);

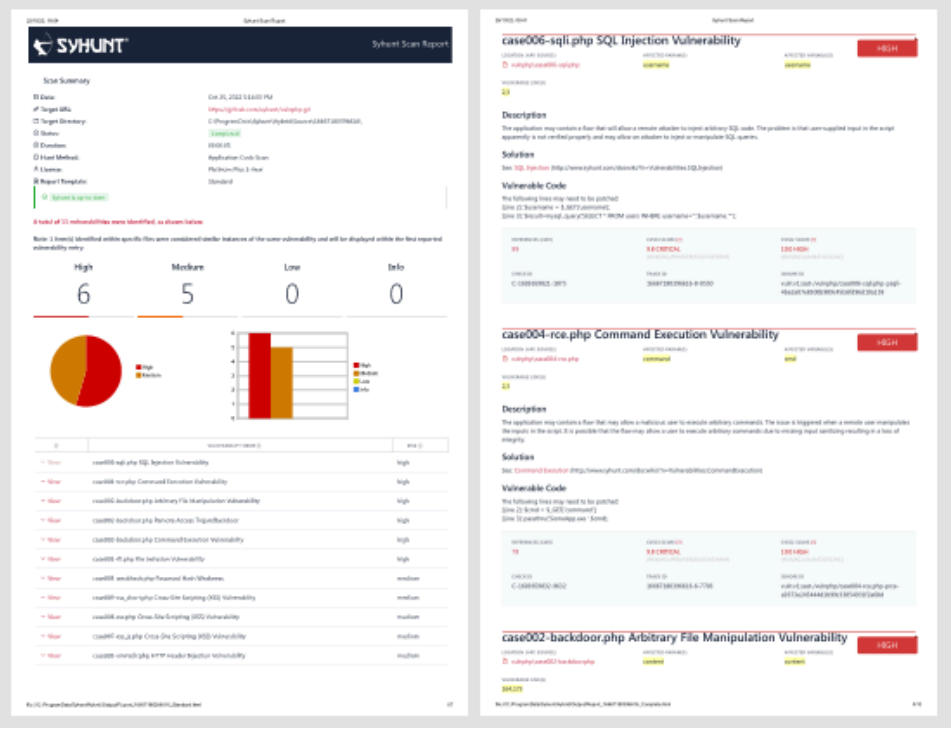
Página 152 de 182

Syhunt Hybrid 7.2

- severidade (Alta, Média, Baixa ou Informativa);
- um ID de verificação único do Syhunt;
- um track ID, único para a vulnerabilidade em cada varredura;
- um guia de correção;
- parâmetros e variáveis afetados;
- números das linhas afetadas;
- um pequeno trecho do código vulnerável, quando aplicável;
- exemplos de código de correção, quando aplicável;
- a requisição, os cabeçalhos e a resposta HTTP, na análise dinâmica;
- dados extraídos, se houver;
- observações do usuário.

F. Recursos de varredura

Relatórios e resultados



O Syhunt apresenta os resultados das varreduras e gera relatórios sobre as vulnerabilidades encontradas. Ele:

- realiza testes de aprovação/reprovação com base na severidade das vulnerabilidades encontradas;
- compara duas varreduras do mesmo alvo ou código, mostrando as vulnerabilidades novas, inalteradas e removidas em um relatório e no console de gerenciamento;
- gera relatórios em inglês, alemão, espanhol, francês, italiano, japonês, coreano e português, com descrições e soluções das vulnerabilidades traduzidas;
- envia os resultados automaticamente por e-mail para um endereço escolhido;
- gera um alerta para cada tipo único de vulnerabilidade encontrado;
- confirma, a partir de vestígios, se uma aplicação foi invadida;
- suporta CVSS 2.0 e 3.0, incluindo a pontuação base e outras métricas, para comunicar a severidade e ajudar a priorizar a resposta;
- gera relatórios em PDF e HTML (para leitura) e em XML, JSON e CSV (para outras ferramentas);
- gera relatórios HTML responsivos para celular, tablet e desktop;
- agrupa instâncias semelhantes de uma vulnerabilidade em um único item;
- ordena as vulnerabilidades por CVSS 3, CVSS 2 (de 0.0 Nenhuma a 10.0 Crítica) ou quatro níveis de severidade (Alta, Média, Baixa, Informativa);

Página 153 de 182

Syhunt Hybrid 7.2

- cria relatórios personalizados com seu próprio título, logotipo, observações e rodapé, e escolha de modelos e seções;
- gera relatórios que abrangem várias sessões selecionadas;
- adiciona automaticamente aos relatórios capturas das páginas autenticadas.

Modelos de relatório

Modelo	Conteúdo
Standard	Relatório gerencial padrão.
Comparison	Compara os resultados de uma varredura com varreduras anteriores, com gráficos e tabelas que deixam as diferenças claras.
Compliance	Relatório de aprovação/reprovação em relação ao OWASP Top 10, ao CWE/SANS Top 25 Most Dangerous Software Errors e ao PCI DSS, em suas versões atuais.
Compliance (Mobile)	Relatório de aprovação/reprovação em relação ao OWASP Mobile Top 10, ao CWE/SANS Top 25 Most Dangerous Software Errors e ao PCI DSS, em suas versões atuais.
Complete	Comparação, conformidade e todos os detalhes técnicos e das vulnerabilidades.

Os relatórios incluem:

- detalhes gerais da varredura: data de início, alvo, status, duração e método de varredura;
- gráficos e estatísticas: total de vulnerabilidades e totais por severidade;
- detalhes das vulnerabilidades: tudo o que é exibido para cada alerta no console de gerenciamento;
- detalhes de cobertura: a estrutura mapeada, listas de formulários, e-mails, arquivos JavaScript e outros recursos encontrados, e as tecnologias e plataformas detectadas;
- a versão e uma breve descrição da licença ativa.

Métodos de varredura

Método	Foco
Application Scan (Dynamic)	Mapeia a estrutura do site e realiza análise passiva e ataques ativos.
Application Code Scan	Todos os tipos de vulnerabilidades no código-fonte.
Application Scan (Server-Side Focused)	Apenas vulnerabilidades do lado do servidor (dinâmica ou de código).
Spider Only	Apenas mapeia a estrutura do site, sem análise passiva nem ataques.
Passive Scan	Mapeia a estrutura do site com análise passiva, sem ataques.
CWE Top 25	O CWE Top 25 Most Dangerous Software Errors.
OWASP Top 10	O OWASP Top 10 Web Application Security Risks.
PHP Top 5	O OWASP PHP Top 5, mas não limitado a PHP.
Fault Injection	Falhas de injeção de dados, como XSS, injeção de SQL, inclusão de arquivos e execução de comandos.
Structure Brute Force	Arquivos de backup comuns, páginas de administração e exposições semelhantes.
Backup Files	Arquivos de backup, ocultos e obsoletos, de forma menos agressiva que o Structure Brute Force.
Complete Penetration Test	Todos os testes dinâmicos, de forma extensa; demorado.
SQL Injection	Injeção de SQL e de NoSQL.
XSS	Cross-site scripting e evasão de filtros anti-XSS.
File Inclusion	Inclusão de arquivo local e remoto (LFI e RFI).
Malicious Content	Malware, backdoors, pontos de entrada ocultos e sinais de intrusão.

Página 154 de 182

Syhunt Hybrid 7.2

Método	Foco
Unvalidated Redirects	Vulnerabilidades de redirecionamento.

Consulte Métodos de Varredura para ver os nomes na CLI e mais detalhes.

Detecção de injeção de SQL

O Syhunt encontra injeção de SQL e de NoSQL por análise dinâmica, usando técnicas inferenciais (baseadas em tempo), in-band (baseadas em erro) e out-of-band, e por análise de código-fonte:

Banco de dados	DAST (inferencial)	DAST (in-band)	SAST	OAST (OOB)
Injeção de NoSQL	✅	✅	✅	N/A
MongoDB	✅	✅	✅	N/A
Injeção de SQL	✅	✅	✅	✅
Firebird/InterBase	N/A	✅	✅	N/A
IBM Db2	N/A	✅	✅	N/A
Informix	N/A	✅	✅	N/A
MariaDB / MySQL	✅	✅	✅	N/A
Microsoft Access	N/A	✅	✅	N/A
Microsoft SQL Server	✅	✅	✅	✅
Oracle	✅	✅	✅	✅
PostgreSQL	✅	✅	✅	N/A
SQLite	✅	✅	✅	N/A
Sybase	✅	✅	✅	N/A

Detecção de injeção e divulgação de código

O Syhunt encontra injeção e divulgação de código por análise dinâmica, usando técnicas inferenciais (baseadas em tempo), in-band (baseadas em impressão) e passivas, e por análise de código-fonte:

Linguagem ou ambiente	DAST (inferencial)	DAST (divulgação)	DAST (in-band)	SAST (injeção)
ASP Classic	✅	✅	✅	✅
ASP.NET	✅	✅	✅	✅
Java	✅	✅	✅	✅
JavaScript	✅	N/A	✅	✅
Lua (NGINX, Apache, CGILua e outros)	✅	✅	✅	✅
Object Pascal (incluindo DWS)	✅	N/A	✅	✅
Perl	✅	✅	✅	✅
PHP	✅	✅	✅	✅
Python	✅	N/A	✅	✅
Ruby	✅	N/A	✅	✅
Server-Side Includes (SSI)	N/A	✅	N/A	N/A

Página 155 de 182

Syhunt Hybrid 7.2

Detecção de injeção de comandos do sistema operacional

O Syhunt encontra injeção de comandos do sistema operacional por análise dinâmica, usando técnicas inferenciais (baseadas em tempo), in-band (baseadas em impressão) e out-of-band, e por análise de código-fonte:

Sistema	DAST (inferencial)	DAST (in-band)	SAST	OAST (OOB)
Unix/Linux	✅	✅	✅	✅
Windows	✅	✅	✅	✅
BSD	✅	✅	✅	✅
Solaris	✅	✅	✅	✅
macOS	✅	✅	✅	✅
iOS	N/A	N/A	✅	N/A
Android	N/A	N/A	✅	N/A

Categorias de vulnerabilidades

O Syhunt detecta os seguintes tipos de vulnerabilidades, fraquezas e exposições em aplicações web e, quando aplicável, em aplicações mobile. Para a lista detalhada de verificações, consulte Verificações de Vulnerabilidades.

- Abuso e uso indevido de APIs
- Manipulação arbitrária de arquivos
- Backdoors (baseados na web)
- Más práticas
- Quebra de autenticação
- Criptografia quebrada
- Estouro de buffer
- Injeção de código, injeção de expression language (EL) e injeção de expressões regulares
- Execução de comandos
- Arquivos e pastas de backup comuns, e backups com extensões comuns ou duplas
- Fraquezas comuns em formulários, incluindo sequestro de formulário de e-mail, campos de preço ocultos, preenchimento automático habilitado e transações de cartão de crédito sem criptografia
- Manipulação de cookies
- Cross-site scripting (XSS), incluindo XSS baseado em DOM, XSS específico de HTML5, filtros de XSS fracos e cross-frame scripting (XFS)
- Cross-site scripting (XSS) em aplicações web com LLM
- Métodos perigosos
- Pontos de entrada de depuração, incluindo parâmetros de depuração ocultos
- Conteúdo padrão
- Negação de serviço (DoS), do lado do cliente e do servidor
- Listagem de diretórios
- Travessia de diretórios (directory traversal)
- Divulgação de caminhos, código-fonte, bancos de dados, senhas, endereços IP internos, tecnologias web e mais
- Falsificação: falsificação de logs, cross-site request forgery (CSRF) e server-side request forgery (SSRF)
- Informações sensíveis fixas no código ou registradas em logs
- Injeção de cabeçalho HTTP e divisão de resposta HTTP (HTTP response splitting)
- Conteúdo impróprio ou malicioso
- Inclusão de arquivo local ou remoto
- Vazamento de informações
- Injeção de LDAP
- Injeção de JSON, XML, XPath e XXE (entidade externa XML)
- Injeção de NoSQL, SQL e HQL
- Injeção de SSI (Server-Side Includes)

Página 156 de 182

Syhunt Hybrid 7.2

- Comunicação insegura
- Algoritmos criptográficos e de hash inseguros
- Armazenamento de dados inseguro: proteção de dados ausente ou insuficiente
- Aleatoriedade insegura
- Uso inseguro de salt
- Cabeçalhos de segurança HTTP ausentes ou fracos
- Configuração de segurança incorreta
- Informações sensíveis no lado do cliente
- Comentários suspeitos no código-fonte e no HTML
- Strings de formato não controladas
- Login sem criptografia
- Redirecionamentos não validados
- Uso de local storage, e dados confidenciais armazenados em local storage
- Hash de senha fraco
- Protocolos fracos

G. Requisitos de sistema

O Syhunt Hybrid roda em Windows, macOS (Apple Silicon) e Linux de 64 bits, e pode analisar aplicações criadas para qualquer plataforma, incluindo Android, iOS, macOS, BSD, Linux, Windows, Solaris e Unix, independentemente da plataforma em que é executado.

Requisito	Detalhes
Memória	4 GB de RAM disponível (8 GB recomendados)[1]
Disco	2 GB de espaço livre em disco[2]
Conexão com a Internet	Recomendada para varreduras de código e dinâmicas, e necessária para alguns recursos (consulte Requisitos de conexão com a Internet)
Sistema operacional (64 bits)	Windows 10 ou 11, ou Windows Server 2016 a 2025 (x64 ou ARM64) macOS Sequoia 15.4.1 ou posterior (Apple Silicon) Linux: Kali 2024.3 ou posterior, Ubuntu Server/Desktop 18 ou posterior, Debian 9 ou posterior Sistemas suportados não oficialmente[3], como as distribuições Linux listadas abaixo
Git	Opcional, para varreduras de repositórios Git: Git no Linux/macOS, Git for Windows no Windows
Java	Java ou Java Headless, no Linux e no macOS
Wine	Wine64 Stable, quando ainda não houver binários nativos para seu sistema ou distribuição
IA	Opcional, para recursos com IA: uma chave de API da OpenAI ou da DeepSeek, ou um modelo de IA local

[1] Não inclui a RAM adicional que pode ser necessária para varreduras simultâneas.
[2] Não inclui o espaço para dados das sessões de varredura, que depende do site ou do código analisado e da frequência das varreduras.
[3] *Suportado não oficialmente* significa que o produto foi testado com sucesso e a instalação foi documentada, mas a Syhunt não oferece suporte técnico para problemas específicos desse sistema. Você pode encontrar problemas de compatibilidade, erros ou bugs; para o melhor desempenho e compatibilidade, use um sistema suportado oficialmente.

Distribuições Linux compatíveis

Status	Distribuição
✅ Suportadas oficialmente	Ubuntu Server/Desktop 18.10 e posteriores Debian 9 e posteriores Kali Linux 2024.3 e posteriores

Página 157 de 182

Syhunt Hybrid 7.2

Status	Distribuição
✅ Suportadas não oficialmente (testadas com sucesso)	Fedora 32 e posteriores CentOS 7.7 e posteriores (Minimal ou Everything) Kali Linux 2019 e posteriores Parrot OS 4.1, 4.7 e posteriores Linux Mint 19.2 e posteriores openSUSE Leap 15.1 e posteriores MX Linux 19.1 e posteriores KDE neon 2020.03 e posteriores Deepin 15.9 Manjaro 19 Arch Linux 2019 e posteriores
❌ Não suportadas	elementary OS 5.1 (testado com sucesso) e 5.0 CentOS 6.1 (testado com sucesso) Solus 4.1 (instável)

Os passos de instalação de cada distribuição estão no guia de instalação para Linux.

H. Glossário

Termo	Definição
Ataques ativos	Análise de segurança que realiza ataques como força bruta, injeção e negação de serviço.
Com IA (AI-powered)	Um recurso, ou extensão de recurso, baseado em inteligência artificial. Disponível apenas quando há uma integração configurada com um sistema de IA suportado (OpenAI, DeepSeek ou um modelo local).
Ativo	Uma URL, arquivo de código-fonte ou repositório que pode ser analisado.
Análise dinâmica aumentada	A combinação de DAST e OAST.
DAST	Dynamic Application Security Testing (teste dinâmico de segurança de aplicações): a ferramenta interage com uma aplicação web em execução para encontrar vulnerabilidades e fraquezas. Também chamado de teste caixa-preta.
DOM	Document Object Model.
FAST	Forensic Application Security Testing (teste forense de segurança de aplicações).
HAST	Hybrid Application Security Testing (teste híbrido de segurança de aplicações): a combinação de SAST e DAST.
Análise híbrida aumentada	A combinação de SAST, DAST e OAST.
Injeção	Envio de dados aos pontos de entrada de uma aplicação e análise da resposta para determinar se seu código é vulnerável.
Lua	Uma linguagem de script leve e extensível, criada no Brasil, projetada para estender aplicações.
MAST	Mobile Application Security Testing (teste de segurança de aplicações mobile): a ferramenta analisa o código-fonte ou o pacote de uma aplicação para encontrar erros de programação e condições que indicam vulnerabilidades.
NVD	National Vulnerability Database, a base de vulnerabilidades do governo dos EUA.
OAST	Out-of-Band Application Security Testing: a ferramenta tenta fazer uma aplicação web se conectar a outros servidores, revelando vulnerabilidades que de outra forma seriam invisíveis.
OOB	Out-of-band.
OSVDB	Open Source Vulnerability Database.
OWASP	Open Worldwide Application Security Project.
Análise passiva	Análise de segurança que encontra vulnerabilidades e exposições sem realizar ataques nem chamar a atenção.

Página 158 de 182

Syhunt Hybrid 7.2

Termo	Definição
PLUS	Um recurso disponível apenas com a licença Hybrid Platinum Plus.
Prova de conceito (PoC)	Um conjunto de ações que demonstra que um produto funciona como esperado.
Regra	Uma opção que habilita ou desabilita uma ou mais verificações de vulnerabilidades.
SAST	Static Application Security Testing (teste estático de segurança de aplicações): a ferramenta analisa o código-fonte de uma aplicação para encontrar erros de programação e condições que indicam vulnerabilidades. Também chamado de teste caixa-branca.
XHR	XMLHttpRequest, uma API para enviar requisições HTTP ou HTTPS a partir de JavaScript.

Página 159 de 182

Syhunt Hybrid 7.2

Preferências

Esta página lista as principais preferências do Syhunt Hybrid que podem ser definidas pela linha de comando. As preferências são **globais** (afetam todas as varreduras e alvos) ou **específicas de um site** (aplicam-se à URL de uma única aplicação web).

> **Observação**
> As alterações não afetam varreduras já em andamento; elas valem para as varreduras iniciadas depois. As preferências definidas com o `scancore` são persistentes e afetam todas as varreduras futuras.

Conteúdo

- Definindo valores pela CLI
- Preferências de site
- Preferências do Dynamic
- Preferências do Code
- Preferências do Forensic
- Outras preferências do Hybrid
- Preferências de rastreadores

Definindo valores pela CLI

Com `scancore -prefset` e `scancore -tracker:set`, defina o valor de acordo com o tipo da preferência:

Tipo	Como definir
boolean	`-v:true` ou `-v:false`
string	`-v:valor`, `-v:"meu valor"` ou `-vsecret` (para senhas)
integer	`-v:1000`
stringlist	`-fromfile:minhalista.lst` para ler os valores de um arquivo; `-v:[string1],[string2],[etc]` para listar várias strings; ou `-v:valor` / `-v:"meu valor"` para uma única linha

Definir um valor do tipo errado gera um erro, o que protege as preferências contra alterações acidentais:

```
scancore -prefset:hybrid.report.company.logo.urldark -v:true
Error: Invalid type (provided boolean, must be string)

scancore -tg:http://127.0.0.1 -prefset:enabled -v:1
Error: Invalid type (provided integer, must be boolean)
```

Alguns detalhes:

- **Listas de strings:** em `-v:[valor1],[valor2],[valor3]`, cada valor entre colchetes vira uma linha separada, e o resultado é armazenado assim:

  ```
  valor1
  valor2
  valor3
  ```

- **Espaços:** coloque entre aspas duplas os valores que contêm espaços, por exemplo `-v:"meu valor"` ou `-v:"[meu valor1],[meu valor2]"`. Não use aspas simples.
- **Segredos:** o `-vsecret` pede o valor duas vezes, mascarando a digitação, para que valores sensíveis, como senhas, sejam confirmados e nunca apareçam na linha de comando.

Página 160 de 182

Syhunt Hybrid 7.2

- **Arquivos:** no Windows, o arquivo informado em `-fromfile` pode estar na pasta atual ou ser indicado com o caminho completo. No Linux, sempre informe o caminho completo, por exemplo `-fromfile:/caminho/para/minhalista.lst`.

Use `-prefprint` para exibir o valor atual de qualquer preferência. Para exemplos de como definir preferências globais e de sites, consulte Linha de Comando: alterando as preferências.

Preferências de site

As preferências específicas de um site são definidas com `scancore -prefset` e o parâmetro `-tg`, que indica o alvo (exemplos). Elas valem apenas para esse alvo, mas para todas as varreduras DAST dele, qualquer que seja a forma de início: o comando `scanurl`, a interface web, a API REST, a API Lua, o módulo PowerShell ou a interface clássica.

Como os alvos são diferenciados:

- **Caminhos diferentes são alvos diferentes.** Por exemplo, `scancore -tg:http://127.0.0.1 -prefset:enabled -v:true` e `scancore -tg:http://127.0.0.1/demo/ -prefset:enabled -v:true` são armazenados separadamente, embora a URL base seja a mesma. Ao executar `scanurl <url>`, são usadas as preferências que correspondem ao caminho dessa URL.
- **Protocolos e portas diferentes são alvos diferentes.** `https://example.com`, `http://example.com` e `http://example.com:8080` têm, cada um, suas próprias preferências.
- **Precisa de vários perfis para a mesma URL?** Acrescente um parâmetro marcador, como `?syhunt_ref=1`, `?syhunt_ref=2` e assim por diante. A aplicação verificada o ignora, mas o Syhunt trata cada variante como um alvo separado.

Preferência	Tipo	Descrição
`enabled`	boolean	**true** (padrão): habilita as preferências de site para a URL alvo. Esta é uma chave mestra. **false**: desabilita todas as preferências de site e usa apenas as configurações padrão.
`name`	string	Nome descritivo do site alvo (por exemplo, Site da minha empresa). É exibido na interface do Syhunt e não afeta as varreduras.
`critlevel`	string	Nível de criticidade. Valores válidos: **undefined** (não definido), **none** (não crítico), **low**, **medium**, **high**, **critical**. É exibido na interface do Syhunt e não afeta as varreduras. Padrão: **undefined**.
`dynamic.augmented.detectoob`	boolean	**true** (padrão): habilita a execução de OAST com o Syhunt Signal (recomendado), permitindo detectar vulnerabilidades out-of-band. **false**: desabilita todos os testes OAST.
`dynamic.augmented.scanclisource`	boolean	**true** (padrão): habilita a execução de SAST no JavaScript do lado do cliente com o Syhunt Code. **false**: desabilita todos os testes SAST dentro do DAST.
`dynamic.advanced.performdos`	boolean	**true** (padrão): habilita os testes de negação de serviço (DoS) contra a aplicação alvo. Ataques DoS buscam interromper o funcionamento normal de uma aplicação explorando vulnerabilidades ou sobrecarregando seus recursos, deixando-a indisponível para usuários legítimos. **false**: desabilita todos os testes de DoS.
`dynamic.advanced.performauthforce`	boolean	**true** (padrão): realiza força bruta de autenticação, testando automaticamente usuários e senhas comuns em todas as áreas da aplicação que pedem credenciais. **false**: desabilita a força bruta de autenticação.
`dynamic.advanced.forcewebrootstruct`	boolean	**true** (padrão): sempre inclui a raiz web na força bruta de estrutura. **false**: não inclui a raiz web. Ou seja, se a URL alvo for `https://example.com/demo/`, a força bruta atingirá `https://example.com/demo/*`, mas não `https://example.com/*`.
`dynamic.advanced.incrementalscan`	string	Varredura incremental. Valores aceitos: **auto** (padrão) ou **disabled**. Desabilitá-la deixa as próximas varreduras mais lentas, pois o scanner não aproveitará os dados das varreduras anteriores.

Página 161 de 182

Syhunt Hybrid 7.2

Tecnologias web

O Syhunt Dynamic pode executar testes adaptados às tecnologias usadas por uma aplicação web, o que acelera os testes. Por padrão, ele detecta essas tecnologias automaticamente; por isso, a maioria destas opções tem o valor padrão `auto`.

Preferência	Tipo	Descrição
`dynamic.webtech.slang`	string	A principal linguagem de script do lado do servidor da aplicação alvo. Conhecê-la antecipadamente permite ao Syhunt otimizar a varredura e executar apenas as verificações relevantes para essa linguagem. Padrão: **auto**.

Valores aceitos para `dynamic.webtech.slang`:

Valor	Significado
`auto`	O Syhunt determina automaticamente a linguagem do alvo usando seus recursos de identificação (fingerprint).
`lang.asp`	A aplicação usa ASP clássico.
`lang.aspx`	A aplicação usa ASP.NET.
`lang.java`	A aplicação usa Java ou JSP (Java Server Pages).
`lang.kotlin`	A aplicação usa Kotlin.
`lang.ssjs`	A aplicação usa JavaScript do lado do servidor, como Node.js ou similar.
`lang.lua`	A aplicação usa Lua ou LuaJIT.
`lang.perl`	A aplicação usa Perl.
`lang.php`	A aplicação usa PHP.
`lang.python`	A aplicação usa Python.
`lang.ruby`	A aplicação usa Ruby ou Ruby on Rails.

Preferência	Tipo	Descrição
`dynamic.webtech.server`	string	O software de servidor web da aplicação alvo. Conhecê-lo antecipadamente permite ao Syhunt otimizar a varredura e executar apenas as verificações relevantes para esse servidor. Padrão: **auto**.

Valores aceitos para `dynamic.webtech.server`:

Valor	Significado
`auto`	O Syhunt determina automaticamente o software de servidor do alvo usando seus recursos de identificação.
`server.apache`	A aplicação usa Apache.
`server.tomcat`	A aplicação usa Apache Tomcat.
`server.iis`	A aplicação usa Microsoft IIS.
`server.nginx`	A aplicação usa NGINX ou OpenResty.

Preferência	Tipo	Descrição
`dynamic.webtech.os`	string	O sistema operacional da aplicação alvo. Conhecê-lo antecipadamente permite ao Syhunt otimizar a varredura e executar apenas as verificações relevantes para esse sistema. Padrão: **auto**.

Valores aceitos para `dynamic.webtech.os`:

Valor	Significado
`auto`	O Syhunt determina automaticamente o sistema operacional do alvo usando seus recursos de identificação.

Página 162 de 182

Valor	Significado
`os.bsd`	A aplicação roda em BSD.
`os.linux`	A aplicação roda em uma distribuição Linux.
`os.solaris`	A aplicação roda em Solaris.
`os.unix`	A aplicação roda em Unix.
`os.windows`	A aplicação roda em Windows.

Preferência	Tipo	Descrição
`dynamic.webtech.datsql`	string	O principal banco de dados SQL ou NoSQL da aplicação alvo. Conhecê-lo antecipadamente permite ao Syhunt otimizar a varredura e executar apenas as verificações relevantes para esse banco. Padrão: **auto**.

Valores aceitos para `dynamic.webtech.datsql`:

Valor	Significado
`auto`	O Syhunt determina automaticamente o banco de dados do alvo usando seus recursos de identificação.
`db.sql.access`	A aplicação usa Microsoft Access.
`db.sql.db2`	A aplicação usa DB2.
`db.sql.informix`	A aplicação usa Informix.
`db.nosql.mongodb`	A aplicação usa MongoDB.
`db.sql.mysql`	A aplicação usa MariaDB ou MySQL.
`db.sql.oracle`	A aplicação usa Oracle.
`db.sql.postgresql`	A aplicação usa PostgreSQL.
`db.sql.sqlserver`	A aplicação usa Microsoft SQL Server.
`db.sql.sqlite`	A aplicação usa SQLite.

Preferência	Tipo	Descrição
`dynamic.webtech.haswaf`	string	A presença de WAF na aplicação alvo. Padrão: **auto**.

Valores aceitos para `dynamic.webtech.haswaf`:

Valor	Significado
`auto`	O Syhunt determina automaticamente se há WAF.
`inactive`	Não há WAF.
`active`	Há WAF.
`ipbypass`	Há WAF, com regra de liberação por IP.

Preferência	Tipo	Descrição
`dynamic.webtech.waf`	string	O fabricante do WAF ativo. Padrão: **auto**.

Valores aceitos para `dynamic.webtech.waf`:

Valor	Significado
`auto`	O Syhunt tenta determinar automaticamente o fabricante do WAF.

Syhunt Hybrid 7.2

Valor	Significado
`waf.akamai`	Akamai
`waf.aws`	AWS
`waf.barracuda`	Barracuda
`waf.cloudflare`	Cloudflare
`waf.f5`	F5
`waf.fortinet`	Fortinet
`waf.imperva`	Imperva
`waf.msazureappgat`	Microsoft Azure Application Gateway
`waf.modsecurity`	ModSecurity
`waf.nginx`	NGINX
`waf.ogasec`	OGASEC
`waf.radware`	Radware
`waf.sucuri`	Sucuri
`waf.unknown`	Outro

Inferência de versões de tecnologias

O Syhunt Dynamic também pode inferir quais tecnologias e versões de componentes um servidor web executa, analisando diversos aspectos da aplicação e de suas respostas. Assim, ele conhece a pilha de tecnologias sem nenhuma informação sua, para testes mais completos.

Preferência	Tipo	Descrição
`dynamic.webtech.guess.enabled`	boolean	**true** (padrão): tenta identificar as versões do software de servidor web usando o Syhunt HunterSense, mesmo com banners e assinaturas ocultos (recomendado). Esta é uma chave mestra. **false**: desabilita a inferência do software de servidor web.
`dynamic.webtech.guess.apache`	boolean	**true** (padrão): identifica a versão do Apache. **false**: desabilita a inferência da versão do Apache.
`dynamic.webtech.guess.nginx`	boolean	**true** (padrão): identifica a versão do NGINX. **false**: desabilita a inferência da versão do NGINX.
`dynamic.webtech.guess.php`	boolean	**true** (padrão): identifica a versão do PHP. **false**: desabilita a inferência da versão do PHP.
`dynamic.webtech.guess.phusion`	boolean	**true** (padrão): identifica a versão do Phusion Passenger. **false**: desabilita a inferência da versão do Phusion Passenger.
`dynamic.webtech.guess.modssl`	boolean	**true** (padrão): identifica a versão do mod_ssl. **false**: desabilita a inferência da versão do mod_ssl.
`dynamic.webtech.guess.openssl`	boolean	**true** (padrão): identifica a versão do OpenSSL. **false**: desabilita a inferência da versão do OpenSSL.

Alcance do rastreamento

Estas opções restringem até onde o rastreador do Syhunt Dynamic explora. Por padrão não há limites de rastreamento, de modo que o scanner pode explorar o site amplamente; ainda assim, ele aplica regras integradas (que a Syhunt chama de *otimizações*) que o guiam pela estrutura e pelo conteúdo do site e aceleram o mapeamento.

Página 164 de 182

Syhunt Hybrid 7.2

Preferência	Tipo	Descrição
`dynamic.crawling.limitdepth`	boolean	**true**: usa um limite de profundidade. Esta é uma chave mestra. **false** (padrão): sem limite de profundidade.
`dynamic.crawling.maxdepth`	integer	Profundidade máxima (exige o limite de profundidade habilitado). Padrão: **0**, que significa sem limite.
`dynamic.crawling.limittobase`	boolean	**true**: limita o rastreamento ao caminho da URL inicial e a URLs especificamente permitidas. Esta é uma chave mestra. **false** (padrão): sem essa limitação. Se você habilitar esta opção e, por exemplo, o alvo for `https://example.com/demo/` e a aplicação tiver links para `https://example.com/test/`, o caminho `/test/` será ignorado. Como os caminhos das aplicações web costumam ser interligados, isso restringiria demais a varredura; por isso, a opção vem desabilitada.
`dynamic.lists.ignore.allowedpaths`	stringlist	Caminhos cujo rastreamento é permitido (por exemplo, /store/login.php). Padrão: lista vazia.

As listas a seguir não exigem nenhuma chave mestra habilitada:

Preferência	Tipo	Descrição
`dynamic.lists.starturls`	stringlist	URLs iniciais adicionais (por exemplo, /url1.php,/url2.php). Por padrão, o Syhunt Dynamic usa como ponto de partida o caminho da URL alvo informada. Esta chave permite incluir caminhos iniciais adicionais. Por exemplo, se o alvo for `https://example.com/demo/` e você também quiser incluir **/demov2/** e **/demov3/**, mesmo que não haja links para eles em **/demo/**, adicioná-los a esta lista garante que também sejam verificados. Padrão: lista vazia.
`dynamic.lists.ignore.urls`	stringlist	Caminhos a ignorar (por exemplo, /about1.php,/about2.php). Padrão: lista vazia.
`dynamic.lists.ignore.logoutpaths`	stringlist	Caminhos de logout a ignorar (por exemplo, /logout.php). Padrão: lista vazia. O Syhunt Dynamic já vem configurado para detectar os caminhos de logout mais comuns.
`dynamic.lists.ignore.form.names`	stringlist	Formulários a ignorar, pelo nome (por exemplo, search1,search2). Padrão: lista vazia.

Autenticação

Estas opções habilitam a autenticação por formulário web, métodos de autenticação de servidor como Basic e Bearer, ou ambos. Consulte Varredura Autenticada.

Preferência	Tipo	Descrição
`dynamic.formauth.type`	string	Tipo de autenticação. Valores aceitos: **None**, **Standard**, **Smart**, **Recorder**, **Selenium-Auto**, **Selenium-JS**, **Basic**, **Bearer**, **Digest** e **NTLM**. O padrão é **Standard**, ou seja, a autenticação por formulário fica habilitada enquanto os campos de usuário e senha não estiverem vazios.
`dynamic.formauth.username`	string	Usuário do formulário. Padrão: vazio.
`dynamic.formauth.password`	string	Senha do formulário. Padrão: vazio.
`dynamic.formauth.token.encrypted`	string	Token para autenticação Bearer. Padrão: vazio.
`dynamic.formauth.script.recorderjson.encrypted`	stringlist	JSON do Recorder para autenticação. Padrão: vazio. Só é usado se o tipo de autenticação for Recorder.

Página 165 de 182

Syhunt Hybrid 7.2

Preferência	Tipo	Descrição
`dynamic.formauth.script.seleniumjs.encrypted`	stringlist	Script Selenium em Node.js para autenticação. Padrão: vazio. Só é usado se o tipo de autenticação for Selenium-JS.
`dynamic.lists.cookies`	stringlist	Cookies de sessão (por exemplo, ck1=valor;ck=valor). Padrão: vazio.
`dynamic.lists.cookiesforced`	stringlist	Cookies forçados (por exemplo, ck1=valor;ck=valor). Padrão: vazio.

Lista de vulnerabilidades ignoradas

Preferência	Tipo	Descrição
`dynamic.lists.ignore.vulnsbyrules`	stringlist	Vulnerabilidades a ignorar (por ID ou regra), por exemplo vuln:v1:dast-my.js-xss-748. Padrão: vazio.

Preferências do Dynamic

As preferências do Dynamic são globais: valem para todas as varreduras DAST, qualquer que seja a forma de início (o comando `scanurl`, a interface web, a API REST, a API Lua, o módulo PowerShell ou a interface clássica).

Preferência	Tipo	Descrição
`dynamic.checks.dos`	boolean	**true** (padrão): realiza testes de negação de serviço (DoS). **false**: desabilita os testes de DoS.

Emulação de navegador

Preferência	Tipo	Descrição
`dynamic.emulation.redirect.autofollowinstarturl`	boolean	**true** (padrão): trata automaticamente redirecionamentos para fora do domínio na URL inicial. **false**: desabilita esse tratamento.
`dynamic.emulation.mode`	string	Modo de emulação de navegador. Opções válidas: **Chrome**, **Edge**, **Firefox**, **MSIE**, **Opera**, **Safari**. Padrão: **Chrome**.
`dynamic.emulation.javascript.execution`	boolean	**true** (padrão): habilita a emulação de JavaScript. **false**: desabilita totalmente a execução e emulação de JavaScript.
`dynamic.emulation.javascript.simuser`	boolean	**true** (padrão): simula a interação do usuário (teclas, cliques do mouse etc.). **false**: desabilita totalmente a simulação de interação do usuário.
`dynamic.emulation.doxhrcalls`	boolean	**true** (padrão): realiza chamadas XHR. **false**: desabilita totalmente o suporte a chamadas XHR.
`dynamic.emulation.javascript.analyzejs`	boolean	**true** (padrão): analisa o código JavaScript. **false**: desabilita a análise de código JS.
`dynamic.emulation.javascript.analyzexhr`	boolean	**true** (padrão): analisa as chamadas XHR. **false**: desabilita a análise de chamadas XHR.
`dynamic.emulation.cookies.accept`	boolean	**true** (padrão): aceita cookies. **false**: desabilita totalmente o suporte a cookies.
`dynamic.emulation.cookies.maxsize`	integer	Tamanho máximo de cookie, em KB. Padrão: **4** (4 KB).
`dynamic.emulation.cookies.maxnumber`	integer	Número máximo de cookies por site. Padrão: **50**.

Página 166 de 182

Preferência	Tipo	Descrição
`dynamic.emulation.autofill.forms`	boolean	**true** (padrão): preenche automaticamente formulários web (nome, endereço, telefone, e-mail etc.). **false**: desabilita o preenchimento automático de formulários.
`dynamic.emulation.autofill.loginforms`	boolean	**true** (padrão): preenche automaticamente formulários de login. **false**: desabilita o preenchimento automático de formulários de login.
`dynamic.emulation.redirect.autofollow`	boolean	**true** (padrão): segue redirecionamentos automaticamente. **false**: não segue redirecionamentos automaticamente.
`dynamic.emulation.intelparser`	boolean	**true** (padrão): usa análise inteligente de HTML (lida com HTML malformado). **false**: desabilita a análise inteligente de HTML.
`dynamic.emulation.usenavbehavior`	boolean	**true** (padrão): usa o comportamento de navegador. **false**: desabilita o comportamento de navegador.
`dynamic.emulation.referer.send`	boolean	**true** (padrão): envia o cabeçalho Referer. **false**: não envia o Referer.
`dynamic.emulation.useragent`	string	User agent. O padrão é o user agent de um navegador Chrome.
`dynamic.emulation.forceuseragent`	boolean	**true** (padrão): força o user agent informado em todas as situações. **false**: não força o user agent informado.

Alcance do rastreamento e tempo limite

Estas configurações globais limitam até onde o rastreador vai e quanto tempo cada varredura pode durar.

Preferência	Tipo	Descrição
`dynamic.crawling.max.linkspersite`	integer	Número máximo de links por servidor. Padrão: **10000**.
`dynamic.crawling.max.linksnumber`	integer	Número máximo de links por página. Padrão: **250**.
`dynamic.crawling.max.urlsize`	integer	Comprimento máximo de URL, em bytes. Padrão: **16384**.
`dynamic.crawling.max.filesize`	integer	Tamanho máximo de resposta, em KB. Padrão: **1024**.
`dynamic.crawling.analyze.robots`	boolean	**true** (padrão): analisa o robots.txt (se disponível). **false**: desabilita a análise do robots.txt.
`dynamic.options.logout.prevent`	boolean	**true** (padrão): impede que o Syhunt Dynamic acesse acidentalmente a página de logout da aplicação durante o rastreamento ou os testes, evitando o encerramento não intencional da sessão. **false**: desabilita os mecanismos de prevenção de logout.
`dynamic.options.logout.detect`	boolean	**true** (padrão): faz login novamente quando a sessão é encerrada prematuramente. **false**: desabilita o novo login.

Preferência	Tipo	Descrição
`syhunt.dynamic.options.timelimit.enabled`	boolean	**false** (padrão): sem limite de tempo. **true**: habilita o limite de tempo. Esta é uma chave mestra.
`syhunt.dynamic.options.timelimit.value`	integer	Duração máxima de uma varredura. Padrão: vazio (sem limite). Formatos aceitos: **1d** (um dia), **3h** (três horas), **2h30m** (duas horas e trinta minutos) ou **50m** (cinquenta minutos). Ao atingir essa duração, a varredura é cancelada automaticamente.

Syhunt Hybrid 7.2

Intervalo entre requisições

Define um intervalo, em milissegundos, entre as requisições. Não recomendamos usá-lo em DAST, que normalmente envolve milhões de requisições: qualquer intervalo pode tornar as varreduras muito mais longas.

Preferência	Tipo	Descrição
`dynamic.options.delay.enabled`	boolean	**false** (padrão): sem intervalo entre requisições. **true**: habilita o intervalo entre requisições. Esta é uma chave mestra.
`dynamic.options.delay.value`	integer	Intervalo, em ms. Padrão: **3000**.
`dynamic.options.delay.userandom`	boolean	**false** (padrão): sem intervalo aleatório. **true**: usa um intervalo aleatório entre as requisições.

Evasão

Preferência	Tipo	Descrição
`dynamic.evasion.detection.honeypot`	boolean	**true** (padrão): detecta se a aplicação alvo é um honeypot. **false**: desabilita a detecção de honeypot.
`dynamic.evasion.detection.waf`	boolean	**true** (padrão): tenta detectar se a aplicação está atrás de um WAF (firewall de aplicação web). **false**: desabilita a detecção de WAF.
`dynamic.evasion.reqdistrand`	boolean	**true**: distribui as requisições aleatoriamente durante o rastreamento (experimental). **false** (padrão): sem distribuição aleatória.
`dynamic.evasion.evadeflt`	boolean	**true** (padrão): usa evasão de filtros. **false**: desabilita a evasão de filtros. Esta é uma chave mestra.
`dynamic.checks.sub.xssev`	boolean	**true** (padrão): técnica de evasão: técnicas comuns de evasão de filtros de XSS. **false**: desabilita essas técnicas.
`dynamic.checks.sub.utf8dev`	boolean	**true** (padrão): técnica de evasão: decodificação UTF-8. **false**: desabilita essa técnica.

Protocolo

Os valores padrão abaixo são os recomendados: eles são otimizados para lidar com o tráfego web de forma eficiente e oferecem o melhor desempenho de varredura.

Preferência	Tipo	Descrição
`dynamic.protocol.version`	string	Versão do protocolo. Aceita: HTTP/1.1 e HTTP/1.0. Padrão (e recomendado): **HTTP/1.1**.
`dynamic.protocol.ssl.type`	string	Tipo de SSL. Aceita: all, SSLv2, SSLv3, TLSv1, TLSv1_1, TLSv1_2, TLSv1_3, SSHv2. Padrão (e recomendado): **all**, que funciona com todos os tipos de SSL suportados em vez de se restringir a um.
`dynamic.protocol.keepalive`	boolean	**true** (padrão): habilita o suporte a Keep-Alive (recomendado). **false**: desabilita o Keep-Alive, o que deixa as varreduras mais lentas, pois o Keep-Alive acelera as conexões.
`dynamic.protocol.usegzip`	boolean	**true** (padrão): habilita o suporte à compressão GZIP (recomendado). **false**: desabilita o GZIP, o que deixa as varreduras mais lentas, pois a compressão acelera as conexões.
`dynamic.protocol.mknocache`	boolean	**true** (padrão): faz requisições sem cache (no-cache). **false**: desabilita as requisições sem cache.

Página 168 de 182

Syhunt Hybrid 7.2

Preferência	Tipo	Descrição
`dynamic.protocol.timeout.value`	integer	Tempo limite, em ms. Padrão: **10000**.
`dynamic.protocol.retries`	integer	Número de novas tentativas após o tempo limite. Padrão: **2**.
`syhunt.dynamic.options.augmented.jndiserver`	string	Servidor JNDIExploit (host sem porta) para detecção de Log4Shell. Padrão: vazio.
`dynamic.lists.protocol.customhdr`	stringlist	Cabeçalhos de requisição personalizados (por exemplo, X-Custom:UmValor). Padrão: vazio.

Lista de vulnerabilidades ignoradas

Cada vulnerabilidade DAST de um relatório termina com seu ID de exclusão, a forma mais simples e recomendada de suprimir um resultado. Para cenários mais amplos, use regras de exclusão.

Preferência	Tipo	Descrição
`dynamic.lists.ignore.vulnsbyrules`	stringlist	Vulnerabilidades a ignorar (por regra ou ID). Padrão: lista vazia.

Preferências do Code

As preferências do Code são globais: valem para todas as varreduras SAST, qualquer que seja a forma de início (o comando `scancode`, SAST dentro do DAST, a interface web, a API REST, a API Lua, o módulo PowerShell ou a interface clássica).

Preferência	Tipo	Descrição
`code.checks.inflt`	boolean	**true** (padrão): analisa a filtragem de entradas no código-fonte da aplicação. Isso é essencial para eliminar falsos positivos, pois verifica o tratamento adequado das entradas do usuário e reduz detecções incorretas. **false**: desabilita a análise de filtragem de entradas.
`code.options.target.tfsver`	string	Compatibilidade com versões do TFS. Aceita: latest, 2017, 2015, 2013, 2012, 2010. Padrão: **latest** (2018 ou posterior).

Lista de vulnerabilidades ignoradas

Cada vulnerabilidade SAST de um relatório termina com seu ID de exclusão, a forma mais simples e recomendada de suprimir um resultado; para cenários mais amplos, use regras de exclusão. No SAST, em vez de `code.lists.ignore.vulnsbyrules`, a prática recomendada é usar um arquivo `.vulnignore` no repositório verificado.

Preferência	Tipo	Descrição
`code.lists.ignore.vulnsbyrules`	stringlist	Vulnerabilidades a ignorar (por regra ou ID). Padrão: lista vazia.

Tempo limite

Preferência	Tipo	Descrição
`syhunt.code.options.timelimit.enabled`	boolean	**false** (padrão): sem limite de tempo. **true**: habilita o limite de tempo. Esta é uma chave mestra.
`syhunt.code.options.timelimit.value`	integer	Duração máxima de uma varredura. Padrão: vazio (sem limite). Formatos aceitos: **1d** (um dia), **3h** (três horas), **2h30m** (duas horas e trinta minutos) ou **50m** (cinquenta minutos). Ao atingir essa duração, a varredura é cancelada automaticamente.

Página 169 de 182

Syhunt Hybrid 7.2

Preferências do Forensic

As preferências do Forensic são globais: valem para todas as varreduras FAST, iniciadas pelo comando `scanlog` ou pela interface clássica.

Preferência	Tipo	Descrição
`insight.lists.ignorelist.paths`	stringlist	Lista de caminhos a ignorar (por exemplo, /robots.txt,robots2.txt). Padrão: lista vazia.
`insight.lists.ignorelist.srcips`	stringlist	Lista de IPs a ignorar (por exemplo, 1.1.1.1,2.2.2.2). Padrão: lista vazia.
`insight.options.logformat`	string	Formato do log. Aceita: auto, common, apache, iis, ncsa, nginx. Padrão: **auto** (detecta o formato automaticamente).
`insight.options.profileatk`	boolean	**true** (padrão): monta o perfil do atacante. **false**: desabilita a criação de perfis de atacantes.
`insight.options.resolveip`	boolean	**true**: resolve os endereços IP dos atacantes (mais lento). **false** (padrão): não resolve os endereços IP (mais rápido).

Outras preferências do Hybrid

Estas preferências globais valem para todas as varreduras, qualquer que seja a forma de início (os comandos `scancode` ou `scanurl`, a interface web, a API REST, a API Lua, o módulo PowerShell ou a interface clássica).

Preferência	Tipo	Descrição
`hybrid.report.defaultlanguage`	string	ID do idioma dos relatórios. Consulte a lista de idiomas disponíveis. Padrão: **EN** (relatórios em inglês).
`hybrid.report.company.logo.urldark`	string	URL do logotipo da organização para fundo escuro (por exemplo, `http://www.mycompany.com/logo.png`). Padrão: vazio.
`hybrid.lists.riskfine`	stringlist	Redefinição de risco. Cada linha deve usar o formato [ID único da verificação=severidade], por exemplo C-1603660271-7241=low. Padrão: lista vazia.
`syhunt.hybrid.advanced.ai.openai.apikey.encrypted`	string	Chave da OpenAI para habilitar os recursos com IA do Syhunt. Padrão: vazio.
`syhunt.hybrid.advanced.ai.options.exrepsolut`	boolean	**true** (padrão): compartilha trechos de código vulnerável para obter exemplos de correção. Exige `ai_reviewconsent: true` em um arquivo `.syhunt-ci.yml` na raiz de cada repositório alvo. **false**: não compartilha trechos de código vulnerável, mesmo que existam arquivos de consentimento.

Preferências de rastreadores

As preferências de rastreadores são definidas com `scancore -tracker:set`, usando `-to` para indicar o rastreador e `-v` para o valor (exemplos). Elas valem apenas para esse rastreador.

Para exemplos de como criar rastreadores e definir suas preferências, consulte Linha de Comando: usando rastreadores de issues.

As chaves aplicáveis dependem do tipo de rastreador: por exemplo, um rastreador de e-mail usa `smtp.targetport`, enquanto um rastreador do GitHub a ignora. As seções após a tabela listam as chaves normalmente necessárias para cada tipo.

Página 170 de 182

Syhunt Hybrid 7.2

Preferência	Tipo	Descrição
`project.name`	string	Nome do projeto (por exemplo, owner/repo). Padrão: vazio.
`auth.username`	string	Usuário. Padrão: vazio.
`auth.password`	string	Senha. Padrão: vazio.
`auth.token.encrypted`	string	Token de autenticação. Pode ser usado para definir um token de acesso pessoal (PAT) de um serviço. Padrão: vazio.
`auth.targethost`	string	Nome do host. Padrão: vazio.
`api.url`	string	URL da API/do servidor. Padrão: vazio.
`custom.fields`	stringlist	Campos personalizados. Padrão: lista vazia.
`custom.labels`	stringlist	Rótulos personalizados. Padrão: lista vazia.
`scan.minseverity`	string	Severidade mínima. Valores aceitos: Info, Low, Medium ou High. Padrão: **Info**.
`notifyonfailonly`	boolean	**true**: notifica apenas quando uma condição de reprovação é atendida. **false** (padrão): sempre notifica (independentemente de condições de aprovação/reprovação).

Chaves por tipo de rastreador

Rastreador	Chaves a definir
Email	`auth.targethost` (host SMTP), `auth.username` e `auth.password` (credenciais), além das chaves de e-mail abaixo.
GitHub	`project.name` (`owner/repo`), `auth.token.encrypted` (token de acesso pessoal) e, opcionalmente, `custom.labels` (por exemplo, `Vulnerability`).
GitLab	`project.name` (`owner/repo`), `api.url` (URL do servidor GitLab), `auth.token.encrypted` (token de acesso pessoal) e, opcionalmente, `custom.labels` .
Jira	`project.name` (projeto Jira), `api.url` (URL do servidor Jira), `auth.username` e `auth.password` e, opcionalmente, `custom.labels` , `custom.fields` e `jira.defitname` .
DefectDojo	`api.url` (URL do servidor), `auth.token.encrypted` (token de API) e, opcionalmente, `scan.minseverity` , `custom.labels` (tags, por exemplo `scan`) e as chaves do DefectDojo abaixo.
Faraday	`api.url` (URL do servidor), `project.name` (workspace; padrão `my_workspace`), `auth.username` e `auth.password` .

E-mail

Preferência	Tipo	Descrição
`smtp.targetport`	integer	Porta SMTP. Padrão: **587**.
`message.from`	string	Endereço do remetente. Padrão: vazio.
`message.tolist`	string	Endereço do destinatário, ou uma lista de endereços separados por vírgula. Padrão: vazio.
`message.subject`	string	Assunto da mensagem. Padrão: vazio.

Jira

Preferência	Tipo	Descrição
`jira.defitname`	string	Tipo de issue padrão. Padrão: **Task**.

Use `custom.fields` para referenciar campos personalizados pelo nome, e não pelo ID. Para um campo personalizado do tipo SelectList, declare-o como `customfield_10100.value=UmValorPersonalizado` .

Página 171 de 182

Syhunt Hybrid 7.2

DefectDojo

Preferência	Tipo	Descrição
`scan.testtitle`	string	Contexto: título do teste. Padrão: vazio.
`scan.engagementname`	string	Contexto: nome do engagement. Padrão: vazio.
`scan.producttypename`	string	Contexto: nome do tipo de produto. Padrão: vazio.
`scan.productname`	string	Contexto: nome do produto. Padrão: vazio.
`scan.active`	boolean	**true** (padrão): marca os resultados como ativos. **false**: marca-os como inativos.
`scan.verified`	boolean	**true** (padrão): marca os resultados como verificados. **false**: marca-os como não verificados.
`scan.skipduplicates`	boolean	**true**: ignora duplicatas. **false** (padrão): não ignora duplicatas.
`scan.closeoldfindings`	boolean	**true**: fecha resultados antigos. **false** (padrão): não fecha resultados antigos.
`scan.pushtojira`	boolean	**true**: envia os resultados ao Jira. **false** (padrão): não envia ao Jira.

Página 172 de 182

Syhunt Hybrid 7.2

Verificações de Vulnerabilidades

> **Aplica-se a:** Syhunt Hybrid 6.9

A base de verificações do Syhunt é resultado de anos de pesquisa e cobre uma variedade extremamente ampla de ameaças à segurança de aplicações web. As linhas em **negrito** são categorias; as linhas seguintes são as verificações individuais de cada categoria.

Para comparar as verificações entre as edições, consulte Comparação de verificações entre edições.

Verificações de segurança de aplicações web

Verificação	CWE
Vulnerabilidades de Autenticação	
Desvio de Autenticação / Quebra de Autenticação por Análise Dinâmica	CWE-287
Desvio de Autenticação / Quebra de Autenticação por Análise de Código-Fonte	CWE-287
Força Bruta Automatizada de Autenticação (por Formulário e HTTP)	
Divulgação de Senha	CWE-311
Login sem Criptografia	CWE-319
Uso Inseguro de Salt por Análise de Código-Fonte	
Criptografia Quebrada	
Algoritmos de Hash Inseguros por Análise de Código-Fonte	
Algoritmos Criptográficos Inseguros por Análise de Código-Fonte	
Aleatoriedade Insegura por Análise de Código-Fonte	
Protocolos Fracos por Análise de Código-Fonte	
Hash de Senha Fraco por Análise de Código-Fonte	
Confirmação de Invasão	
Confirmação de Invasão por Análise Dinâmica	
Confirmação de Invasão por Análise de Código-Fonte	
Confirmação de Invasão por Análise de Logs do Servidor Web	
Descoberta de Parâmetros de Depuração Ocultos por Análise Dinâmica	
Descoberta de Parâmetros de Depuração Ocultos por Análise de Código-Fonte	
Descoberta e Injeção de Parâmetros de Depuração Ocultos por Análise Híbrida	
Detecção de Conteúdo Impróprio	
Detecção de Conteúdo Malicioso	
Detecção de Backdoors Web por Análise Dinâmica	
Detecção de Backdoors Web por Análise de Código-Fonte	
Execução de Comandos	CWE-78
Execução de Comandos por Análise Dinâmica	

Página 173 de 182

Syhunt Hybrid 7.2

Verificação	CWE
Execução de Comandos por Análise de Código-Fonte	
Cross-Site Scripting (XSS)	CWE-79
Cross-Site Scripting (XSS) por Análise Dinâmica	
Cross-Site Scripting (XSS) por Análise de Código-Fonte	
Desvio de Filtro/Validação Fraca de XSS por Análise Dinâmica	
Desvio de Filtro/Validação Fraca de XSS por Análise de Código-Fonte	
XSS Baseado em DOM por Análise de Código-Fonte	
XSS Específico de HTML5	
Cross-Site Request Forgery (XSRF) por Análise de Código-Fonte	
Inclusão de Arquivos	CWE-98
Inclusão de Arquivo Local (LFI) por Análise Dinâmica	
Inclusão de Arquivo Remoto (RFI) por Análise Dinâmica	
Inclusão de Arquivo Local (LFI) por Análise de Código-Fonte	
Inclusão de Arquivo Remoto (RFI) por Análise de Código-Fonte	
Manipulação de Arquivos por Análise de Código-Fonte	
Injeção de NoSQL	
Injeção de NoSQL por Análise Dinâmica (Baseada em Erro)	
Injeção de NoSQL por Análise Dinâmica (Baseada em Tempo)	
Injeção de NoSQL por Análise de Código-Fonte	
Injeção de NoSQL no MongoDB	
Injeção de SQL	CWE-89
Injeção de SQL por Análise Dinâmica (Baseada em Erro)	
Injeção de SQL por Análise Dinâmica (Às Cegas)	
Injeção de SQL por Análise Dinâmica (Baseada em Tempo)	
Injeção de SQL por Análise de Código-Fonte	
Injeção de SQL por Análise de Código-Fonte (HQL)	
Injeção de Código	CWE-94
Injeção de Código por Análise Dinâmica (Baseada em Impressão)	
Injeção de Código por Análise Dinâmica (Baseada em Tempo)	
Injeção de Código por Análise de Código-Fonte	
Injeção de Código em ASP Classic	
Injeção de Código em ASP.NET	
Injeção de Código em Java/JSP	
Injeção de Código em Lua (Nginx, Apache, CGI-Lua etc.)	
Injeção de Código em Perl	
Injeção de Código em PHP	

Página 174 de 182

Syhunt Hybrid 7.2

Verificação	CWE
Injeção de Código em Python	
Injeção de Código em Ruby	
Injeção de Código em JavaScript do Lado do Servidor	
Divulgação de Código-Fonte	CWE-540
Divulgação de Código-Fonte por Injeção	
Divulgação de Código-Fonte por Análise de Conteúdo	
Divulgação de Código-Fonte (ASP Classic)	
Divulgação de Código-Fonte (ASP.NET)	
Divulgação de Código-Fonte (Java/JSP)	
Divulgação de Código-Fonte (Lua)	
Divulgação de Código-Fonte (Perl)	
Divulgação de Código-Fonte (PHP)	
Divulgação de Código-Fonte (SSI)	
Verificação de Extensões	
Verificação de Extensão Dupla	
Extensões de Backup Comuns	
Força Bruta de Estrutura	
Páginas de Administração	
Arquivos de Backup Comuns	
Pastas de Backup Comuns	
Divulgação de Banco de Dados	
Arquivos Antigos/de Backup	CWE-530
Fraquezas Comuns em Formulários	
Sequestro de Formulário de E-mail	
Campo de Preço Oculto em Formulário	
Preenchimento Automático Habilitado (em campos sensíveis)	
Transação de Cartão de Crédito sem Criptografia	
Negação de Serviço (DoS)	CWE-730
Negação de Serviço do Lado do Cliente	
Negação de Serviço por Injeção	
Negação de Serviço por Análise de Código-Fonte	
Estouro de Buffer	CWE-120
Vulnerabilidades Diversas de Divulgação	
Exposições Comuns	
Informações Sensíveis Fixas no Código por Análise de Código-Fonte	
Registro de Informações Sensíveis em Log por Análise de Código-Fonte	

Página 175 de 182

Syhunt Hybrid 7.2

Verificação	CWE
Uso de Local Storage por Análise de Código-Fonte	
Dados Sensíveis Armazenados em Local Storage por Análise de Código-Fonte	
Informações Sensíveis no Lado do Cliente por Análise de Código-Fonte	
Divulgação de Informações por Injeção	CWE-200
Divulgação de Informações por Análise de Conteúdo	CWE-200
Divulgação de Informações por Análise de Código-Fonte	CWE-497
Divulgação de Endereço IP Interno	CWE-200
Divulgação de Caminhos por Injeção	CWE-211
Divulgação de Caminhos por Análise de Conteúdo	CWE-211
Listagem de Diretórios	CWE-548
Divulgação de Tecnologias Web	
Comentários HTML Suspeitos	
Falsificação de Logs	CWE-117
Falsificação de Logs por Análise de Código-Fonte	
Injeção de XPath	CWE-91
Injeção de XPath por Análise Dinâmica	
Injeção de XPath por Análise de Código-Fonte	
Injeção de LDAP	CWE-90
Injeção de LDAP por Análise Dinâmica	
Injeção de LDAP por Análise de Código-Fonte	
Redirecionamentos Não Validados	CWE-601
Redirecionamentos Não Validados por Análise Dinâmica	
Redirecionamentos Não Validados por Análise de Código-Fonte	
Injeção de Cabeçalho CRLF	CWE-93
Injeção de Cabeçalho CRLF por Análise Dinâmica	
Injeção de Cabeçalho CRLF por Análise de Código-Fonte	
Injeção de Expression Language (EL)	CWE-917
Manipulação de Cookies	
Cross Frame Scripting	CWE-352
Métodos Perigosos	CWE-749
Conteúdo Padrão	CWE-276
Travessia de Diretórios (Directory Traversal)	CWE-22
Injeção de Cabeçalho HTTP por Análise de Código-Fonte	
Vulnerabilidades Específicas de Servidores IIS, iPlanet e Outros	
Server-Side Request Forgery (SSRF) por Análise Dinâmica	CWE-918
Server-Side Request Forgery (SSRF) por Análise de Código-Fonte	CWE-918

Página 176 de 182

Syhunt Hybrid 7.2

Verificação	CWE
Injeção de Server-Side Includes (SSI)	
Client-Side Request Forgery (CSRF) por Análise de Código-Fonte	
Injeção de XML por Análise Dinâmica	CWE-661
Injeção de XML por Análise de Código-Fonte	CWE-661
Injeção de Entidade Externa XML (XXE)	CWE-827
Injeção de Entidade Externa XML (XXE) por Análise Dinâmica	
Injeção de Entidade Externa XML (XXE) por Análise de Código-Fonte	
Configuração de Segurança Incorreta por Análise de Código-Fonte	
Aplicações Vulneráveis Conhecidas	
Scripts Desatualizados e Vulneráveis por Análise de Código-Fonte	
Aplicações Vulneráveis Conhecidas (Apache Struts)	
Aplicações Vulneráveis Conhecidas (ASP Classic)	
Aplicações Vulneráveis Conhecidas (ASP.NET)	
Aplicações Vulneráveis Conhecidas (ColdFusion)	
Aplicações Vulneráveis Conhecidas (Dynamic HTML)	
Aplicações Vulneráveis Conhecidas (Flash)	
Aplicações Vulneráveis Conhecidas (Java / JSP)	
Aplicações Vulneráveis Conhecidas (Perl)	
Aplicações Vulneráveis Conhecidas (Python)	
Aplicações Vulneráveis Conhecidas (Ruby)	
Aplicações Vulneráveis Conhecidas (SSI)	
Aplicações Vulneráveis Conhecidas (IIS)	

Linguagens suportadas

- Análise dinâmica (DAST): consulte Varredura Dinâmica.
- Análise de código-fonte (SAST): consulte Varredura de Código-Fonte.
- Mobile (MAST): consulte Varredura de Código-Fonte.

Bancos de dados suportados (detecção de injeção de SQL)

- Access
- DB2
- dbx
- Firebird/InterBase
- FrontBase
- Informix
- Ingres
- MariaDB / MySQL
- MaxDB
- mSQL
- Oracle

Página 177 de 182

Syhunt Hybrid 7.2

- Ovrimos
- PostgreSQL
- SQL Server
- SQLite
- Swish
- Sybase

Página 178 de 182

Syhunt Hybrid 7.2

Suporte

Página 179 de 182

Syhunt Hybrid 7.2

Como Relatar Bugs e Obter Suporte

> **Aplica-se a:** Syhunt Hybrid 6.8.20

Conteúdo

- Varreduras DAST longas
- Como relatar um falso positivo
- Como relatar um falso negativo
- Como relatar um problema de rastreamento
- Como relatar outros bugs
- Como criar um log de depuração
- Como abrir um ticket de suporte

Varreduras DAST longas

A varredura de um site com muito conteúdo gerado dinamicamente pode levar bastante tempo. Consulte Ajuste de Desempenho para entender o motivo e como resolver.

Como relatar um falso positivo

Um *falso positivo* é um resultado que, após revisão manual, se mostra não ser uma vulnerabilidade ou fraqueza. Para relatá-lo e permitir a correção:

1. Descreva cada falso positivo com o máximo de detalhes possível.
2. Crie um log de depuração.
3. Abra um ticket de suporte e anexe os arquivos de depuração e sua descrição.

Como relatar um falso negativo

Um *falso negativo* é uma vulnerabilidade que existe no site ou código alvo, mas não foi detectada. Para relatá-lo e permitir a correção:

1. Descreva o tipo de vulnerabilidade (por exemplo, injeção de SQL).
2. **Varreduras de código:** inclua os **trechos relevantes** do código-fonte que contêm a vulnerabilidade não detectada.
3. **Varreduras dinâmicas:** descreva o caso com o máximo de detalhes possível, incluindo:
    - a URL afetada, se houver;
    - os passos usados para confirmar a vulnerabilidade;
    - para falhas de injeção, o parâmetro afetado, os dados injetados e uma amostra da resposta da aplicação.
4. Crie um log de depuração.
5. Abra um ticket de suporte e anexe os arquivos de depuração, sua descrição e eventuais trechos de código.

Como relatar um problema de rastreamento

1. Informe se a URL alvo está protegida por alguma barreira, como autenticação por formulário de login. Isso nos ajuda a analisar o procedimento, a configuração ou o método usado e a sugerir ou fazer ajustes na ferramenta.
2. Se a URL que não está sendo rastreada corretamente estiver acessível, envie-a para nós com credenciais de teste (se necessário), para que possamos testá-la diretamente em nosso laboratório.
3. Se isso não for possível, crie um log de depuração.
4. Abra um ticket de suporte e anexe os arquivos de depuração e uma descrição do problema.

Página 180 de 182

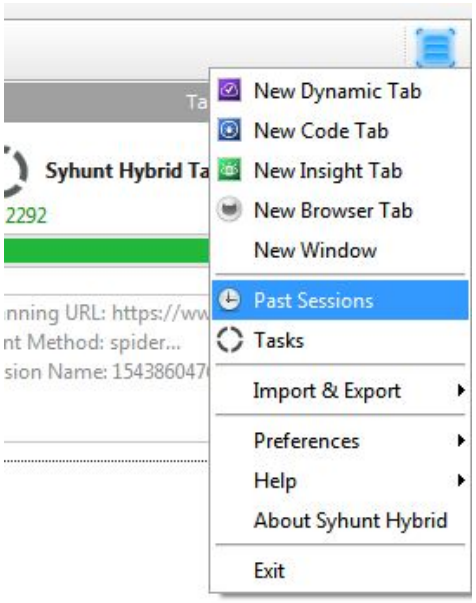
Syhunt Hybrid 7.2

Como relatar outros bugs

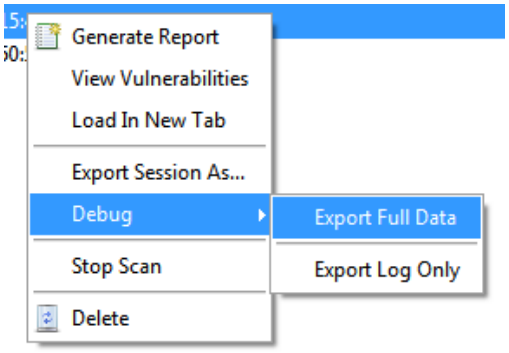
1. Descreva o problema com o máximo de detalhes possível, para que possamos investigá-lo e reproduzi-lo em nosso laboratório.
2. Faça uma captura de tela do problema.
3. Crie um log de depuração.
4. Abra um ticket de suporte e anexe os arquivos de depuração, sua descrição e a captura de tela.

Como criar um log de depuração

1. Abra a tela de sessões em **Menu** → **Past Sessions**.



2. Clique com o botão direito na sessão que deseja depurar e escolha **Debug** → **Export Full Data** para salvar o arquivo.



Isso cria um arquivo de depuração com a extensão `.sse`. Envie-o à Syhunt para análise.

Como abrir um ticket de suporte

Um ticket de suporte é a melhor forma de relatar um bug ou pedir ajuda.

1. Acesse a área de tickets de suporte e clique em **Submit Ticket**.
2. Se ainda não tiver uma conta, clique em **Register**, informe um endereço de e-mail válido e seu nome e clique em **Register**. Verifique seu e-mail para obter os dados de acesso, volte à página de login, entre e clique em **Submit Ticket**.
3. Preencha o formulário: escreva sua mensagem, anexe os arquivos necessários e clique em **Send**.

Página 181 de 182

Syhunt Hybrid 7.2

Um agente de suporte da Syhunt analisará o problema, e você receberá por e-mail uma cópia de cada resposta. O prazo esperado de resposta é de 1 a 2 dias úteis. Você pode acompanhar o status dos seus tickets em My Tickets e responder por e-mail ou pela área de tickets.

Página 182 de 182