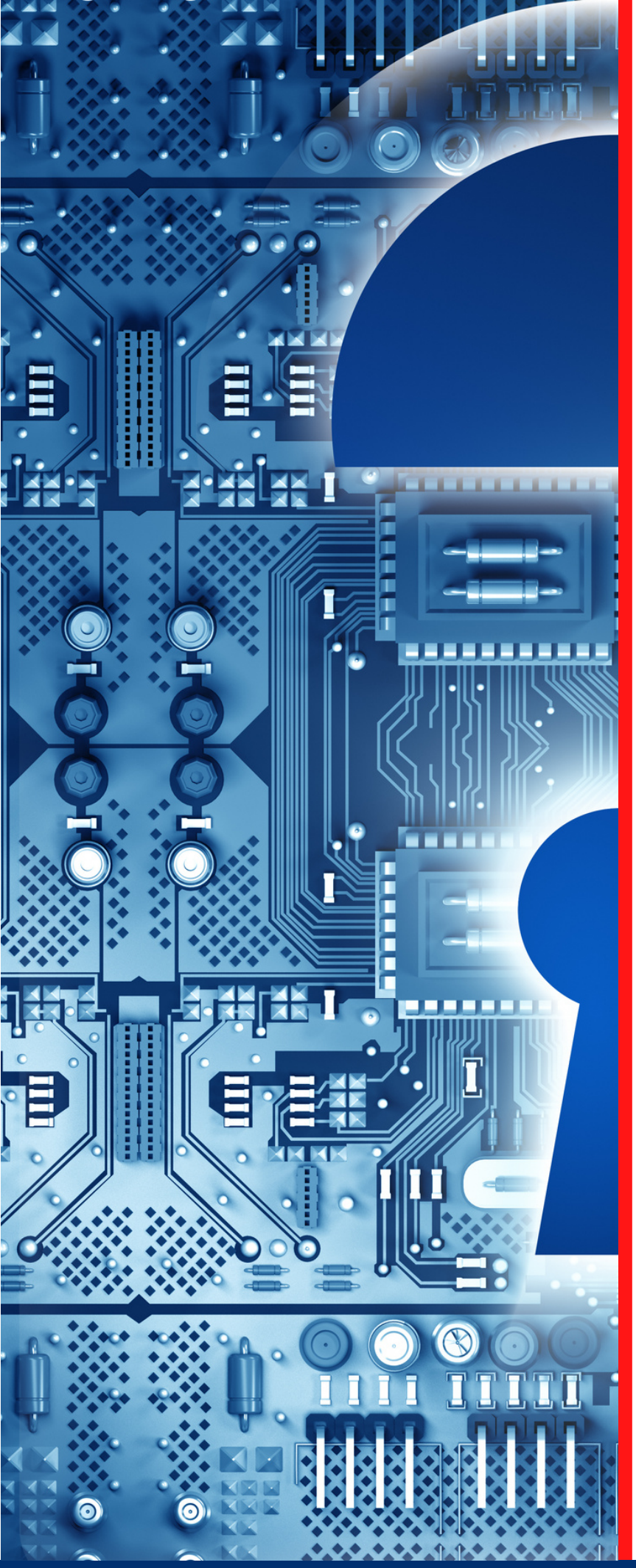


OSIC

ORIENTAÇÃO DE SEGURANÇA DA INFORMAÇÃO E CIBERNÉTICA

05/2023

SCANNING

Textos: João Alberto Muniz Gaspar
Diagramação: Douglas Rocha de Oliveira
Produção: Secretaria de Segurança da Informação e Cibernética

Espaço cibernético inclusivo, seguro, estável, acessível e pacífico.

TLP:CLEAR

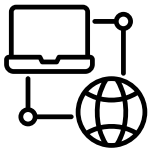
Abuso de Sítio Eletrônico de Governo - *SCANNING*

Conforme estatística do Centro de Prevenção, Tratamento e Resposta a Incidentes Cibernéticos de Governo – CTIR Gov, os ataques de varredura (***scanning***) ocuparam, em 2022, o 3º lugar no *ranking* de incidentes cibernéticos em *sites* de governo. No Brasil, no mesmo ano, quase 60% dos incidentes reportados ao Centro de Estudos, Resposta e Tratamento de Incidentes de Segurança no Brasil (CERT.br) foram relativos a ataques de varredura.

O ***scanning*** é um conjunto de procedimentos que usa técnicas de reconhecimento complexas e ofensivas para identificar *hosts*, portas abertas e serviços ativos. Dessa forma, em muitos casos, identifica o sistema operacional e a arquitetura do sistema de destino, observa vulnerabilidades e ameaças na rede. Em seguida, essas informações são utilizadas pelo atacante para criar um perfil da organização de destino.

De uma forma geral, o *scanning* é a fase inicial, denominada preparação, de ataque a uma rede. Durante esta fase, o *hacker* verifica possíveis alvos em busca de vulnerabilidades na infraestrutura, bem como vulnerabilidades ainda desconhecidas, ataques *zero-day*.[1]

Frequentemente, os *hackers* usam ferramentas automatizadas para o ataque de varredura, por vezes as mesmas que as equipes de segurança utilizam para mapear seus sistemas ou identificar falhas de segurança.



Importante entender que o processo de varredura é realizado contra qualquer *host* disponível, ou seja, qualquer computador ou dispositivo conectado. Observa-se que os *hosts* recebem pelo menos um endereço de rede.

Ataque de varredura padrão

Um ataque de varredura padrão é composto por várias etapas. Geralmente, as etapas são as que seguem abaixo.

1) Enumeração de destino

Nesta fase, o *hacker* define especificadores de *host* (conjunto de DNS de *host*, endereços IP, notações de rede, entre outros) que devem ser testados e utiliza ferramentas para resolver esses especificadores em uma lista de endereços IPv4 ou IPv6 para verificação. Esta fase é importante para todos os procedimentos posteriores de varredura.

2) descoberta de *hosts*

Com os resultados obtidos na enumeração de destino, verifica-se os alvos na rede que estão ativos e, portanto, merecem investigação detalhada. Diversas técnicas podem ser utilizadas para a descoberta de *host*, variando de solicitações *Address Resolution Protocol*[2] (ARP *Request*) rápidas a combinações elaboradas de *Transmission Control Protocol*[3] (TCP-*Request*), *Internet Control Message Protocol*[4] (ICMP *Echo Request*) e outros tipos de testes.

3) Identificação do sistema operacional do *host*

Geralmente, essa etapa se inicia com a varredura de portas (*port scanning*), que consiste na execução de comandos de sondagem numa lista pré-definida de portas. As respostas a essas sondagens são usadas para classificar as portas remotas em estados como aberto, fechado ou filtrado. Um estado aberto indica que a aplicação na máquina-alvo está em execução. Um estado classificado como filtrado indica que um *firewall* está bloqueando a porta, o que não permite determinar se a porta está aberta ou fechada. Um estado fechado é indicação de que a aplicação não está escutando na porta.

Se qualquer porta for determinada como aberta, o *hacker* usará diferentes técnicas para determinar qual o sistema operacional está sendo executado no *host*. Geralmente, essa identificação é realizada por meio do envio de diversos comandos de sondagem através das portas abertas e pela comparação de qualquer resposta obtida com bases de dados de serviços de assinatura, de forma a identificar não só o sistema operacional em execução, mas também a sua versão.

Identificar qual sistema operacional é executado no *host* de destino é muito útil para o *hacker*. Conhecer o tipo específico de sistema operacional permite ao *hacker* determinar, por extensão, sua configuração padrão. Identificar a versão instalada permite ao *hacker* explorar vulnerabilidades conhecidas e não mitigadas ou corrigidas, especialmente se o servidor não está utilizando uma versão atualizada do sistema operacional. Além disso, o simples fato de conhecer o sistema operacional em uso permite ao *hacker* concentrar suas atividades na exploração das soluções e aplicações mais comumente utilizadas com o sistema operacional identificado, a fim de usar suas vulnerabilidades para acessar arquivos de amostra e explorar contas de usuário padrão.

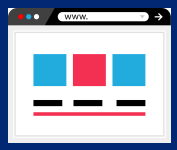
4) Identificação da infraestrutura do *host*

Informações adicionais importantes sobre a infraestrutura dos *hosts* de destino podem ser coletadas usando técnicas de sondagem, como revelação de caminho, *Directory Traversal* e execução remota, o que pode permitir o mapeamento de todo o *site* e sua fonte. O *hacker* pode completar a base de conhecimento de infraestrutura identificando tipos de servidor de banco de dados, tipos de infraestrutura de conteúdo e assim por diante.

Depois que o *hacker* analisa a infraestrutura, todo o aplicativo pode ser verificado. A verificação do aplicativo fornece um mapa de todo o *site*, incluindo todas as páginas, parâmetros usados por páginas dinâmicas, *cookies* usados pelo *site* e fluxo de transações. Essas informações levam o invasor a entender a autenticação, a autorização, a lógica e os mecanismos transacionais do aplicativo. Esse conjunto de informações fornece a base de uma estratégia para atacar o local de destino.

5) Varredura de vulnerabilidades

De posse das informações sobre os sistemas operacionais em uso, da infraestrutura dos *hosts* de destino e do mapa do *site*, o atacante inicia uma varredura de vulnerabilidades, geralmente pelo uso de *bots* maliciosos que investigam locais de conteúdo, caminhos e nomes de arquivos conhecidos e desconhecidos em busca de pontos fracos de segurança, como, por exemplo, sistemas de gerenciamento de conteúdo (CMS, na sigla em inglês) e componentes vulneráveis.



Vale ressaltar que, depois que os adversários identificam as vulnerabilidades de segurança, eles podem definir quais os melhores alvos a serem atacados e qual a melhor forma de realizar o ataque, sendo que os mais comuns incluem a negação de serviço, a instalação de conteúdo malicioso, o controle de conta, fraudes e a exploração de informações confidenciais, entre outros.

Métodos de proteção ante ataques de varredura

Estabelecer configurações que privilegiem a segurança da infraestrutura do *site* e dos dispositivos de rede.

Usar sistemas de detecção de invasões (IDS, na sigla em inglês) e de prevenção de intrusão (IPS, na sigla em inglês) na rede para detectar assinaturas de ataques de varredura e prevení-los.

Aplicar correção de sistemas e componentes, particularmente de segurança, assim que o fabricante lançar uma atualização.

Realizar varreduras de vulnerabilidades e testes de penetração para identificar falhas de segurança.

Desativar ou substituir tecnologias ou recursos obsoletos ou que não estão sendo utilizados ou que são considerados inseguros.

Embora essas medidas de segurança possam reduzir o problema, elas não são completamente eficientes contra as gerações mais recentes de *bots* sofisticados. Entretanto, como a detecção em tempo real e a resposta a ataques são fundamentais para mitigar ataques de varredura, é importante utilizar soluções capazes de identificar as principais operações executadas nesses ataques, em especial, as seguintes:

1) geração de erros usando endereços de rede (URLs, na sigla em inglês) não existentes: esse tipo de atividade só pode ser detectado por soluções que aprendem quais URLs são permitidos por cada aplicativo específico. Os sistemas de detecção e prevenção de intrusão que não são orientados a aplicativos da *web* não implementam esse recurso;

2) injeção de valores de parâmetros longos: para detectar valores de parâmetro longos, a solução deve conhecer as restrições de comprimento em cada parâmetro, o que requer restrições de parâmetros de aprendizagem. Entretanto, os sistemas de detecção e prevenção de intrusão que não são orientados a aplicativos da *web* não implementam esse recurso;

3) acesso (tentativa e sucesso) a partes não autorizadas do ambiente: para detectar tanto tentativas como o acesso a áreas não autorizadas, a solução deve obter conhecimento sobre quais partes do aplicativo são autorizadas e quais não são. Observa-se que somente produtos que incluem recursos de aprendizado podem obter esse conhecimento;

4) adição e remoção de parâmetros: para detectar esse comportamento, a solução deve entender quais parâmetros são usados com cada URL específico e quais são obrigatórios. Destaca-se que os sistemas de detecção e prevenção de intrusão que não são orientados a aplicativos da *web* não implementam esse recurso.

O Departamento de Segurança da Informação e Cibernética (DSIC) recomenda, ainda, que:

- seja elaborado um atualizado plano de continuidade de negócios; e
- qualquer usuário que constate um incidente cibernético, informe imediatamente a Equipe de Prevenção, Tratamento e Resposta a Incidentes Cibernéticos (ETIR) da instituição.

O DSIC solicita ainda que propostas de temas, sugestões ou outras contribuições sejam encaminhadas ao e-mail educa.si@presidencia.gov.br para fomentar futuras publicações da OSIC.

Por fim, outros conceitos podem ser verificados no glossário do DSIC disponível em:

https://www.gov.br/gsi/pt-br/composicao/SSIC/dsic/glossario-de-seguranca-da-informacao-1

TLP:CLEAR

Informações complementares

1. **Ataques *zero day*:** um ataque *zero-day* é uma vulnerabilidade desconhecida que é explorada por atacantes antes que qualquer ação possa ser tomada para corrigir ou proteger contra ela. O termo "*zero-day*" refere-se ao fato de que não há "dias zero" para corrigir o problema antes que ele seja explorado.

2. ***Address Resolution Protocol(ARP)* :** é um protocolo de camada de enlace de dados usado para mapear endereços IP para endereços físicos, como endereços MAC em uma rede local. Ele é necessário para que os dispositivos possam se comunicar uns com os outros na rede.

3. ***Transmission Control Protocol:*** é um protocolo de transporte usado na camada de transporte do Modelo de Referência de Protocolo de *Internet* (TCP / IP). O objetivo principal do TCP é garantir a entrega confiável e ordenada de dados entre aplicativos em dispositivos diferentes na rede.

4. ***Internet Control Message Protocol (ICMP)*:** é um protocolo de nível de *internet* usado para enviar mensagens de controle e indicadores de status em uma rede. Ele é usado para comunicações de erro, diagnósticos e informações de status entre dispositivos na rede, como *routers* e computadores.

5. ***Directory Traversal*:** é uma técnica de invasão de segurança de sistemas de computador que permite a um invasor acessar arquivos e pastas em um sistema remoto que normalmente não seriam acessíveis. O ataque usa a manipulação dos caminhos de arquivos especificados em uma solicitação de arquivo para acessar arquivos e pastas além da área acessível autorizada.