TLP: CLEAR

BOLETIM DE SEGURANÇA

Ataques de injeção de código utilizando chaves MachineKey do ASP.NET

heimdall
security research

Acesse a nossa nova comunidade através do WhatsApp!

Heimdall Security Research

Acesse boletins diários sobre agentes de ameaças, *malwares*, indicadores de comprometimentos, TTPs e outras informações no *site* da ISH.

Boletins de Segurança – Heimdall

SUMÁRIO

LISTA DE TABELAS

LISTA DE FIGURAS

1 INTRODUÇÃO EXECUTIVA

Este relatório de segurança, desenvolvido pela equipe de **Inteligência de Ameaças da ISH, Heimdall**, tem como objetivo proporcionar uma compreensão aprofundada e um dimensionamento preciso das ameaças cibernéticas identificadas. O documento está estruturado em três níveis de abordagem: ***Estratégico***, ***Tático*** e ***Operacional***, garantindo uma visão completa e integrada das ameaças e ações recomendadas.

2 ESTRATÉGICO

2.1 SEGMENTO DE MERCADO

Os alvos potencialmente afetados por essa ameaça que será descrita neste relatório, incluem:

- *Setor financeiro*
- *Comércio eletrônico*
- *Saúde*
- *Setor público*
- *Educação*
- *Tecnologia e Desenvolvimento de Software*

2.2 IMPACTO FINANCEIRO POTENCIAL

- *Interrupção de serviços essenciais*
- *Roubo de dados sensíveis*
- *Custos associados à remediação e investigação de incidentes*
- *Danos à reputação da marca*

2.3 OBJETIVO DA AMEAÇA

O principal objetivo dos atacantes é obter execução remota de código nos servidores IIS alvo, permitindo o controle total do sistema comprometido. Isso possibilita a execução de comandos maliciosos, extração de dados e implantação de ferramentas de pós-exploração, como o framework Godzilla.

3 TÁTICO

3.1 INFORMAÇÕES SOBRE A AMEAÇA

Em dezembro de 2024, a Microsoft identificou atividades maliciosas envolvendo um ator de ameaça não atribuído que utilizou uma chave MachineKey do ASP.NET disponível publicamente para injetar código malicioso e implantar o framework de pós-exploração Godzilla. A investigação revelou que desenvolvedores, ao incorporarem chaves MachineKey de fontes públicas em seus aplicativos, inadvertidamente expuseram seus sistemas a ataques de injeção de código ViewState. A Microsoft identificou mais de 3.000 chaves divulgadas publicamente que podem ser exploradas nesses tipos de ataques.

O ViewState é um método que o ASP.NET usa para preservar o estado da página e dos controles entre postbacks. Os dados do ViewState são armazenados em um campo oculto na página e codificados em Base64. Para proteger o ViewState contra adulteração e divulgação de informações, o framework ASP.NET utiliza chaves MachineKey: ValidationKey e DecryptionKey. Se essas chaves forem comprometidas ou tornadas acessíveis a atores maliciosos, eles podem criar um ViewState malicioso usando as chaves roubadas e enviá-lo ao site via uma solicitação POST. Quando a solicitação é processada pelo Runtime do ASP.NET no servidor alvo, o ViewState é descriptografado e validado com sucesso porque as chaves corretas são usadas. O código malicioso é então carregado na memória do processo de trabalho e executado, fornecendo ao ator de ameaça capacidades de execução remota de código no servidor IIS alvo.

3.2 OPERAÇÃO E CAPACIDADE DA AMEAÇA

O ator de ameaça conduziu o ataque de injeção de código ViewState aproveitando uma chave MachineKey publicamente conhecida. O payload malicioso do ViewState carregou de forma reflexiva o assembly.dll, um framework de pós-exploração Godzilla, seguido por módulos de plugin. A funcionalidade do Godzilla inclui a execução de comandos maliciosos, injeção de shellcode em processos e mais. A ameaça permite que atacantes executem código arbitrário nos servidores **IIS comprometidos**, proporcionando controle total sobre o sistema afetado. Isso inclui a capacidade de:

Execução Remota de Código (RCE)

- Os atacantes podem injetar código malicioso em servidores IIS que utilizam ASP.NET, permitindo a execução remota de comandos sem necessidade de autenticação.

Persistência no sistema

- Com o controle do servidor, os invasores podem instalar backdoors, criar contas privilegiadas e garantir acesso contínuo para futuras explorações.

Escalonamento de privilégios

- A partir do acesso inicial, é possível elevar privilégios no sistema comprometido, obtendo controle administrativo e movendo-se lateralmente dentro da infraestrutura da vítima.

Exfiltração de dados sensíveis

- Os atacantes podem capturar credenciais, tokens de autenticação, dados financeiros e informações de clientes, comprometendo a segurança e a conformidade regulatória.

Desdobramento de ataques secundários

- O comprometimento inicial pode ser usado para disseminar malware, lançar ataques de ransomware, instalar keyloggers ou explorar outras vulnerabilidades na rede da organização.

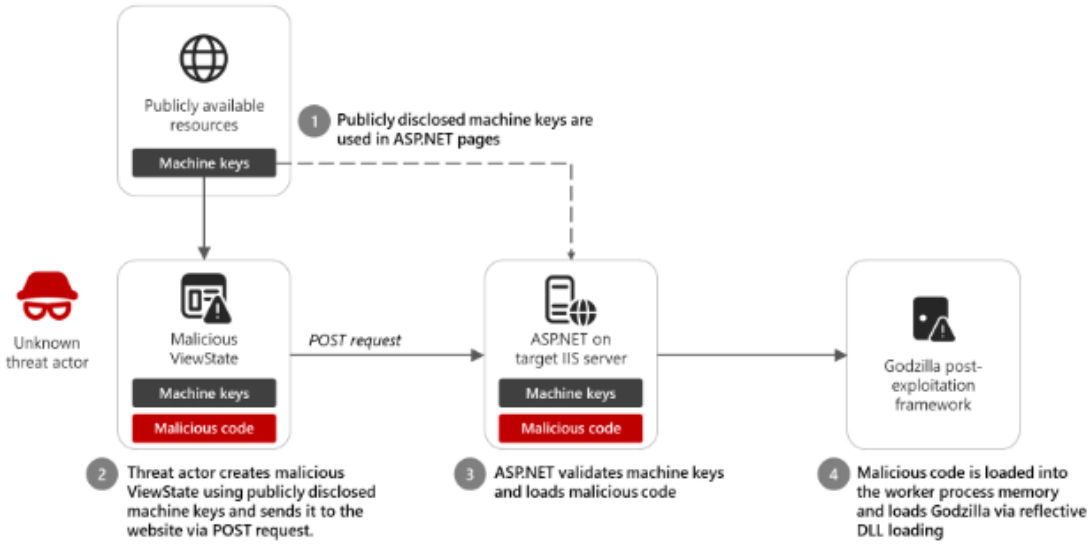

Figura 1 – Cadeia de ataque de injeção de código do ViewState que leva ao Godzilla.

Conforme já mencionado, os agentes da ameaça exploraram chaves MachineKey do ASP.NET divulgadas publicamente como um vetor para realizar injeção de código malicioso em servidores IIS. Nessas campanhas, os atacantes utilizaram chaves estáticas publicadas em repositórios online e documentação pública para gerar ViewStates maliciosos, permitindo a execução remota de código (RCE) sem necessidade de autenticação. O principal objetivo era comprometer aplicações web baseadas em ASP.NET, injetando cargas maliciosas, incluindo o framework de pós-exploração Godzilla, que oferece capacidades avançadas de controle e persistência no ambiente comprometido. Atores de ameaça não identificados já utilizaram essa técnica em ataques observados em dezembro de

2024, demonstrando a viabilidade dessa abordagem para furtar dados sensíveis, manipular informações em servidores web e manter acesso persistente aos sistemas-alvo.

A ameaça representada pela exploração de chaves MachineKey do ASP.NET requer atenção por parte das organizações, especialmente aquelas que utilizam servidores IIS para aplicações web críticas. Suas capacidades avançadas de execução remota de código (RCE), persistência no ambiente comprometido e implantação de cargas maliciosas, como o framework Godzilla, tornam essa técnica um vetor de ataque preocupante. Além disso, a exploração de ViewStates manipulados pode permitir roubo de credenciais, comprometimento de dados sensíveis e movimentação lateral na rede. As organizações devem adotar medidas proativas de segurança, como rotação periódica das chaves MachineKey, implementação de monitoramento de logs para detectar atividades suspeitas e reforço das políticas de segurança em aplicações ASP.NET.

3.3 TABELA MITRE ATT&CK

Este tópico apresenta as Táticas, Técnicas e Procedimentos (TTPs) identificados nesta ameaça, conforme o framework MITRE ATT&CK, oferecendo uma visão tática detalhada sobre o comportamento do adversário. O objetivo é permitir o mapeamento das técnicas utilizadas pelos atacantes, facilitando a implementação de contramedidas eficazes e o aprimoramento das defesas de segurança.

Tática	Técnica	Detalhes
Initial Access	T1190 - Exploit Public-Facing Application	Os adversários exploram vulnerabilidades em aplicativos voltados para a internet para obter acesso inicial.
Execution	T1059.001 - Command and Scripting Interpreter: PowerShell	Os adversários podem abusar do PowerShell para executar comandos e scripts maliciosos.
Persistence	T1505.003 - Web Shell	Os adversários podem instalar shells web para manter acesso persistente aos sistemas comprometidos.
Defense Evasion	T1027 - Obfuscated Files or Information	Os adversários podem ofuscar arquivos ou informações para evitar a detecção.
Command and Control	T1071.001 - Application Layer Protocol: Web Protocols	Os adversários podem usar protocolos da camada de aplicação, como HTTP ou HTTPS, para se comunicar com sistemas comprometidos.

Tabela 1 – Tabela MITRE ATT&CK.

4 RECOMENDAÇÕES

Abaixo são elencadas pela ISH, medidas que poderão ser adotadas visando a mitigação da referida *ameaça*, como por exemplo:

Rotacionar regularmente as chaves MachineKey

- Substitua periodicamente as chaves MachineKey do ASP.NET para evitar o uso de chaves comprometidas. Evite reutilizar chaves disponíveis publicamente.

Monitorar logs de aplicações e servidores

- Implemente soluções de SIEM (Security Information and Event Management) para identificar atividades suspeitas, como ViewStates manipulados e tentativas de execução remota de código.

Evitar o uso de chaves públicas ou pré-configuradas

- Nunca utilize chaves MachineKey que foram copiadas de repositórios públicos ou documentações, pois podem estar comprometidas. Gere chaves únicas e seguras para cada aplicação.

Aplicar regras de firewall e WAF

- Utilize Web Application Firewalls (WAFs) para bloquear cargas maliciosas e solicitações suspeitas. Reforce regras de firewall para restringir tráfego suspeito e desnecessário.

Atualizar regularmente o framework ASP.NET

- Mantenha o .NET Framework e o ASP.NET sempre atualizados com os últimos patches de segurança para mitigar vulnerabilidades conhecidas exploradas por atacantes.

Implementar autenticação e controle de acesso

- Utilize MFA (Autenticação Multifator) e restringir privilégios de usuários e serviços para minimizar o impacto de acessos não autorizados.

Revisar e fortalecer a segurança do ViewState

- Habilite a opção ViewStateUserKey para proteger contra ataques de replay e assine digitalmente o ViewState para evitar manipulação de código malicioso.

5 REFERÊNCIAS

- **Heimdall *by* ISH Tecnologia**
- **Microsoft**
- **Bleepingcomputer**

6 AUTORES

- **Leonardo Oliveira**

heimdall
security research
A DIVISION OF ISH