SIMULAÇÃO E ESTUDO DE MALWARES

Autor: Guilherme Silva

Curso: Santander - Cibersegurança 2025

Instituição: Dio.me

Contato: Guilhermedasilvadossantos2006@gmail.com

Sumário

1. INTRODUÇÃO

Neste documento, apresento de forma didática o processo de criação e funcionamento de dois tipos de malware simulados: Ransomware e Keylogger. O objetivo deste projeto foi compreender, na prática, técnicas de captura de teclas utilizadas por keyloggers, métodos de criptografia e sequestro de arquivos

empregados por ransomware, além de analisar e propor medidas de defesa aplicáveis em ambientes corporativos.

Todos os testes e experimentos foram realizados em uma máquina virtual isolada, garantindo um ambiente seguro e controlado.

2. OBJETIVOS

- Compreender o funcionamento básico de keyloggers em nível de usuário.
- Compreender a criptografia e estrutura do ransomware.
- Documentar testes e resultados em ambiente controlado.
- Propor medidas de mitigação e práticas de segurança.

3. CONTEXTO ÉTICO E LEGAL

Todos os experimentos foram conduzidos com consentimento e em ambiente controlado. A execução deste tipo de código em máquinas de terceiros é ilegal e antiética sem autorização expressa.

4. FERRAMENTAS E AMBIENTE

- VM: Windows 10 — VirtualBox (snapshot antes de cada teste)
- Python 3.10 (uso para prototipação)
- Bibliotecas:
  - pynput (apenas conceitual)
  - cryptography (Fernet — para explicar cifragem de logs local)

5. RANSOMWARE SIMULADO

5.1 Arquitetura e Fluxo

1. Gerar chave de criptografia.
2. Salvar chave em arquivo local.
3. Carregar chave quando necessário.
4. Percorrer uma pasta e identificar arquivos.
5. Criptografar arquivos e sobrescrever conteúdo.
6. Criar arquivo “LEIA ISSO.txt” com mensagem fictícia.

5.2 Explicação Geral do Código

O script segue seis etapas principais: - Criação da chave de criptografia usando Fernet. - Salvamento da chave no arquivo chave.key. - Carregamento da chave para operação. - Varredura do diretório de testes. - Criptografia dos arquivos encontrados. - Criação de uma mensagem de resgate fictícia para demonstração.

5.3 Explicação das Funções

- gerar_chave() – Cria e salva a chave em chave.key.
- carregar_chave() – Retorna a chave salva.
- criptografar_arquivo(arquivo, chave) – Abre, lê, criptografa e sobrescreve o arquivo.
- encontrar_arquivos(diretorio) – Lista arquivos, ignorando o script e a chave.
- criar_mensagem_resgate() – Escreve a mensagem de aviso em “LEIA ISSO.txt”.
- main() – Coordena toda a execução.

5.4 Pseudocódigo do Ransomware

```
gerar chave
guardar chave em "chave.key"

carregar chave

listar arquivos da pasta "test_file"

para cada arquivo:
    abrir arquivo
    ler conteúdo
    criptografar usando a chave
    sobrescrever arquivo com dados criptografados

criar "LEIA ISSO.txt" com mensagem
mostrar mensagem no terminal
```

Observação: uso estritamente educacional. Não incluir rotas de rede ou funções de exfiltração em repositórios públicos.

6. **KEYLOGGER SIMULADO**

6.1 Arquitetura e Fluxo

1. O programa inicia um listener de teclado usando pynput.
2. Cada tecla pressionada é capturada e registrada em memória (variável log).
3. Um temporizador (Timer) executa automaticamente a função de envio de e-mail a cada 60 segundos.
4. Caso o log contenha dados, o conteúdo é enviado para um e-mail previamente configurado.
5. O log é limpo após cada envio para evitar duplicação.
6. O keylogger continua rodando em segundo plano enquanto a captura estiver ativa.
7. O script pode ser executado de forma oculta ao ser salvo como .pyw.

6.2 Explicação Geral do Código

Este keylogger possui duas funções principais: - on_press(key): registra teclas pressionadas no buffer log. - enviar_email(): envia periodicamente o conteúdo do log por e-mail.

O código utiliza: - smtplib para envio SMTP via Gmail. - MIMEText para formatar o corpo do e-mail. - Timer para agendar o envio automático a cada 60 segundos. - Uma variável global log que acumula os dados capturados.

Assim como o modelo básico, teclas normais são adicionadas diretamente ao log, enquanto teclas especiais possuem tratamento individual.

Execução em Segundo Plano (ocultar janela)

Para que o programa seja executado sem abrir janela, basta alterar a extensão de:

keylogger_email.py → keylogger_email.pyw

No Windows, arquivos .pyw executam em segundo plano de forma discreta.

6.3 Explicação das Funções

on_press(key) – Captura as teclas pressionadas. - Adiciona caracteres normais com key.char. - Converte teclas especiais como espaço e enter. - Ignora teclas como Shift, Ctrl, Alt. - Registra BACKSPACE como [<].

enviar_email() – Responsável por: - Verificar se há dados no log. - Montar um e-mail com MIMEText. - Autenticar no servidor SMTP. - Enviar a mensagem contendo tudo que foi digitado no período. - Limpar o log após o envio. - Agendar novamente o envio (Timer(60, enviar_email)).

keyboard.Listener – Mantém o keylogger ativo indefinidamente.

6.4 Pseudocódigo do Keylogger (com envio por e-mail)

```
iniciar variável log vazia
carregar configurações de e-mail (origem, destino e senha)

função enviar_email():
    se log não estiver vazio:
        criar mensagem de e-mail com conteúdo do log
        conectar ao servidor SMTP
        autenticar
        enviar mensagem
        limpar o log

    agendar enviar_email novamente para rodar em 60 segundos

função on_press(tecla):
    tentar:
        adicionar tecla.char ao log
    senão:
        se tecla é espaço: adicionar " "
        se tecla é enter: adicionar "
"
        se tecla é backspace: adicionar "[<]"
        caso contrário: ignorar

iniciar listener chamando on_press
iniciar enviar_email
manter execução enquanto o listener estiver ativo

(opcional) salvar arquivo como .pyw para rodar em segundo plano
```

7. MEDIDAS DE DEFESA (Resumo prático)

- Instalar e manter EDR/antivírus atualizados
- Restringir privilégios de instalação para usuários padrão
- Usar MFA e gerenciadores de senhas; evitar digitar senhas em sistemas potencialmente inseguros
- Monitoramento de processos e integridade de arquivos
- Treinamento de conscientização contra phishing

8. CONCLUSÃO

Este estudo permitiu analisar e compreender em profundidade o funcionamento de dois tipos de malware educativos: Ransomware e Keylogger, incluindo sua arquitetura interna, fluxo de execução, técnicas de operação e possíveis métodos de exfiltração. A implementação do ransomware demonstrou como a criptografia pode ser usada para sequestro de arquivos, reforçando a importância de políticas de backup, controle de acesso e monitoramento de integridade.

O keylogger, por sua vez, apresentou uma visão prática sobre captura de eventos do teclado e geração de logs, além de uma segunda versão mais avançada que realiza exfiltração periódica por e-mail, simulando técnicas utilizadas por ameaças reais para envio de dados sensíveis. A análise das funções, fluxo e comportamento permitiu compreender com clareza como essas ferramentas operam em ambiente controlado.

A realização do projeto evidencia a importância de práticas de segurança ofensiva e defensiva, ressaltando que o conhecimento desses mecanismos é essencial para reforçar a proteção de sistemas corporativos. Ao final, o estudo reforça que a melhor defesa é o entendimento detalhado da ameaça, acompanhado de boas políticas de uso, monitoramento ativo e treinamento contínuo de usuários e equipes técnicas.

9. ANEXOS

Repositório no GitHub — Malware Simulado (Keylogger e Ransomware)
https://github.com/Griff-OFC/DIO_Script-s/tree/main/MALWARES