Documentação de Estudo sobre Keyloggers

1. Introdução

Keyloggers são ferramentas que registram tudo digitado por um usuário em um computador. Essas ferramentas exitem em varias formas (software, hardware, extensões de navegador etc) e podem ser usadas tanto para fins legais quanto para fins ilegais.

1.1 Diferença entre conceito e ferramenta

Um **keylogger como conceito** é apenas a ideia neutra de registrar teclas digitadas, podendo ser usado em auditoria, pesquisa ou ferramentas legítimas. Já um **keylogger como ferramenta de ataque** é a aplicação maliciosa desse conceito, criada para roubar dados de forma oculta, com funções de espionagem, persistência e evasão de segurança.

1.2 Relevância no contexto de segurança atual

A relevância dos keyloggers no contexto de segurança atual permanece alta porque eles continuam sendo uma das formas mais simples e eficazes de roubo de credenciais. Mesmo com avanços em autenticação e melhorias nos sistemas operacionais, a captura de digitação ainda contorna grande parte das proteções tradicionais, já que atua no ponto mais vulnerável: o próprio usuário. Além disso, keyloggers são frequentemente incorporados a malwares modernos, como stealers, RATs e trojans bancários, tornando-se parte de campanhas complexas de espionagem digital. O crescimento do home office, do acesso remoto e do uso de múltiplos dispositivos também ampliou a superfície de ataque, aumentando as possibilidades de infecção. Por esses motivos, keyloggers continuam sendo uma ameaça relevante e atual, exigindo medidas de proteção baseadas em autenticação multifator, monitoramento comportamental, antivírus atualizado e práticas seguras de uso por parte dos usuários e das empresas.

2. Evolução e Contexto Histórico

A ideia de registrar discretamente a entrada de dados existe muito antes da computação moderna, antes mesmo até da criação dos computadores pessoais. A ferramenta de keylogger que conhecemos hoje são somente uma parte de sua evolução, que surgiu ainda na era da espionagem física e mecânica.

Como praticamente tudo na tecnologia, a necessidade desse tipo de captura de dados originou do meio militar. Em um contexto de Guerra Fria, keylogging surgiu como uma técnica de inteligência física e eletromecânica, usada principalmente pela KGB e pela CIA. Na tentativa de interceptação de informações confidenciais de ambientes altamente sensíveis.

O interessante desse período está na forma de captura desses dados. Para nos citarmos estamos falando de uma época aonde grande parte desses dados eram digitados em máquinas de escrever. Assim, sensores eletromagnéticos eram escondidos para captura desses sinais que cada tecla emitia ao ser pressionada. Os sinais eram gravados e depois analisados por especialistas que decodificavam a sequência para reconstruir a informação digitada. Essa tecnologia não se limitava a somente esse método, mas outros como microtransmissores embutidos na estrutura de máquinas, captura de som acústico etc, também eram populares.

2.1 A Era dos Computadores Pessoais

A próxima grande fase da computação histórica é com certeza a era dos computadores pessoais. Que com sua popularização na década de 80, o conceito de keylogging deixou de ser exclusivamente físico e passou a assumir uma dimensão digital.

Mas foi com sistemas como Windows 95, 98 e XP que keyloggers de **software** se popularizaram, graças a APIs que permitiam interceptar eventos de teclado diretamente no sistema operacional. Técnicas como *hooking*, monitoração do buffer e injeção de DLL tornaram possível capturar tudo o que o usuário digitava.

No fim dos anos 90 e início dos 2000, keyloggers passaram a integrar **Trojans e RATs** como SubSeven, Back Orifice e NetBus, abrindo o caminho para ataques mais sofisticados. Essa fase marcou a consolidação do keylogger como componente essencial de malware voltado ao roubo de credenciais, criando a base técnica para as ameaças modernas que surgiriam com a internet em larga escala.

2.2 A Era da Internet

Com a expansão da internet a partir dos anos 2000, o keylogging deixou de ser um recurso local e passou a operar em **escala global**. Keyloggers começaram a ser incorporados a malwares distribuídos pela web — como trojans bancários, botnets e RATs modernos — permitindo que atacantes capturassem credenciais e enviassem os dados automaticamente pela rede. Nessa fase surgiram também técnicas mais avançadas, como **browser injection**, **web skimmers** (ex.: ataques Magecart), **scripts maliciosos em páginas comprometidas** e keyloggers embutidos em extensões de navegador.

A era da internet transformou o keylogging em um ataque remoto, automatizado e massivo: não era mais necessário ter acesso físico ao dispositivo. Ao mesmo tempo, surgiram variantes mais sofisticadas atuando em **nível de kernel**, **drivers**, **hypervisors** ou utilizando **side channels**, como captura acústica ou eletromagnética. Essa fase consolidou o keylogger como ferramenta central do cibercrime moderno, impulsionando também a evolução das técnicas de detecção, proteção de navegador e segurança bancária.

2.3 Keylogging Moderno

O keylogging contemporâneo representa uma evolução significativa em relação às formas primitivas de captura de teclas do passado. Em vez de atuar como

uma ferramenta isolada, hoje ele se integra a um ecossistema mais amplo de malwares complexos, geralmente presentes em trojans, botnets, RATs e info-stealers. Esses keyloggers modernos não se limitam a registrar teclas digitadas: eles coletam credenciais, formulários, tokens de autenticação, dados bancários e comportamentos de uso, operando em múltiplas camadas do sistema. Essa maturidade tecnológica tornou o keylogger uma arma silenciosa, furtiva e extremamente eficiente em operações de espionagem e fraude.

Nos navegadores, o keylogging assumiu novas formas. Scripts maliciosos injetados em páginas de checkout, extensões adulteradas e ataques do tipo form grabbing capturam dados antes mesmo de serem criptografados pelo TLS. Esse cenário é amplamente explorado por grupos como os associados ao Magecart, especializados em skimming digital. Em paralelo, os sistemas operacionais também passaram a conviver com keyloggers avançados capazes de operar tanto em modo de usuário, por meio de APIs legítimas e injeção de DLLs, quanto no nível de kernel, utilizando drivers maliciosos que se escondem profundamente na estrutura interna do sistema para evitar detecção.

Técnicas ainda mais sofisticadas surgiram com avanços na pesquisa acadêmica, especialmente na área de ataques por canais colaterais. Métodos que utilizam padrões acústicos do teclado, vibração captada por sensores, variação eletromagnética ou até distorções no sinal Wi-Fi demonstram que o keylogging não depende mais, necessariamente, de software malicioso instalado na máquina-alvo. Da mesma forma, os dispositivos móveis ampliaram o alcance desse tipo de espionagem: permissões abusivas em Android, sobreposições de tela usadas por trojans bancários e spywares avançados em iOS mostram que o ecossistema mobile também se tornou um terreno fértil para captura invisível de dados.

No campo físico, o keylogging moderno também encontrou novas formas de expressão. Dispositivos USB que se passam por teclados, microcontroladores ocultos e adaptadores modificados conseguem registrar o que é digitado sem levantar suspeitas. Somando-se a isso, malwares atuais utilizam técnicas de evasão de alto nível — como ofuscação polimórfica, execução fileless, anti-debug, anti-sandbox e comunicação criptografada — tornando sua presença ainda mais difícil de identificar por ferramentas tradicionais.

Diante desse cenário, a defesa contra keylogging moderno exige uma abordagem multifacetada. Tecnologias como EDR com monitoramento em nível de kernel, hardening de navegadores, restrição de extensões, políticas de

privilégio mínimo, inspeção de tráfego anômalo e validação contínua da integridade de scripts são essenciais. No ambiente corporativo, a combinação entre controles técnicos, cultura de segurança e arquitetura Zero-Trust se torna indispensável para mitigar riscos.

Assim, o keylogging atual não é mais apenas um mecanismo simples de espionagem, mas um componente sofisticado de ataques multivetoriais. Ele reflete a evolução do cibercrime, que explora interfaces, sistemas, redes, navegadores e até o ambiente físico para obter informações sensíveis. Entender sua complexidade é fundamental para desenhar defesas modernas capazes de acompanhar a velocidade das ameaças digitais.

3. Classificação Técnica dos Keyloggers

3.1 Camada de Operação

Os keyloggers podem atuar em diferentes níveis da arquitetura de um sistema, explorando desde componentes físicos até mecanismos de software de alto nível. Cada camada oferece um tipo de acesso distinto ao fluxo de entrada do teclado e, por isso, representa diferentes graus de complexidade, furtividade e impacto.

Hardware:

Nesta camada, o keylogging ocorre por meio de dispositivos físicos conectados entre o teclado e o computador, ou embutidos internamente no próprio periférico. Eles capturam sinais elétricos das teclas antes mesmo de chegarem ao sistema operacional, o que torna esse método extremamente difícil de detectar por software. É um dos tipos mais discretos e eficazes em ambientes sensíveis.

Firmware:

Keyloggers atuando no firmware manipulam o código embarcado em dispositivos como teclados, UEFI ou controladores USB. Como operam abaixo do sistema operacional, conseguem registrar teclas sem serem monitorados por antivírus ou ferramentas tradicionais. São sofisticados, persistentes e geralmente associados a ataques direcionados.

Kernel:

No nível de kernel, o keylogger intercepta drivers de entrada ou funções internas do sistema operacional responsáveis pelo processamento das teclas. Essa abordagem permite alta furtividade e amplo controle, já que o kernel tem

acesso privilegiado. Keyloggers nesse nível costumam estar acoplados a rootkits.

Userland:

Aqui, o keylogging ocorre em nível de usuário, por meio de APIs fornecidas pelo próprio sistema operacional, como *SetWindowsHookEx* no Windows ou eventos X11 em sistemas Linux antigos. São mais fáceis de implementar, porém mais suscetíveis a detecção, já que dependem de processos comuns.

Browser:

Keyloggers na camada do navegador capturam dados digitados em páginas web. Podem atuar via extensões maliciosas, scripts injetados, ataques de phishing ou keylogging baseado em formulários. Essa camada é amplamente explorada por stealers modernos, principalmente visando credenciais de redes sociais e bancos.

Rede:

Embora não capturem diretamente as teclas, keyloggers de rede interceptam dados antes ou depois da digitação ser enviada a um servidor. Isso inclui sniffers em redes inseguras, ataques MITM e manipulação de tráfego para capturar credenciais transmitidas sem criptografia.

Side-channels:

Nesta camada, a captura não ocorre pelo fluxo digital normal, mas sim por efeitos colaterais do hardware, como ruído eletromagnético, padrões de energia, vibração ou variações temporais. Alguns estudos mostram que é possível inferir teclas pressionadas analisando sons do teclado ou consumo elétrico, tornando esse método extremamente avançado e difícil de mitigar.

3.2 Métodos de Capturas

Os keyloggers podem empregar diferentes métodos para interceptar a entrada do teclado, variando em nível de complexidade, profundidade de acesso ao sistema e grau de furtividade. Cada método explora um ponto específico da cadeia de processamento de entrada, desde funções de alto nível fornecidas pelo sistema operacional até mecanismos que atuam antes da criptografia de dados enviados para a web.

API Hooking (SetWindowsHookEx, GetAsyncKeyState):

Esse é um dos métodos mais comuns e acessíveis, utilizado por keyloggers que operam no espaço do usuário. Ele consiste em interceptar funções da API do sistema que tratam eventos de teclado, como *SetWindowsHookEx* no Windows, ou fazer consultas contínuas do estado das teclas usando *GetAsyncKeyState*. É simples de implementar e eficaz, porém mais suscetível à detecção por antivírus.

Injeção de DLL:

Nesse método, o invasor injeta uma biblioteca dinâmica (DLL) dentro de processos de interesse, normalmente navegadores ou aplicações sensíveis. Uma vez dentro do processo, a DLL pode acessar APIs internas e capturar eventos de entrada de forma mais furtiva e precisa. É frequentemente usado como etapa intermediária em malwares mais sofisticados.

Drivers maliciosos:

Os keyloggers baseados em drivers operam em nível de kernel, interceptando diretamente o driver de teclado ou implementando um driver próprio para monitorar dados de entrada. Por atuarem abaixo da camada de usuário, são difíceis de identificar e podem contornar restrições de segurança impostas pelo sistema operacional. São típicos de rootkits e ataques altamente direcionados.

Logging de buffer de teclado:

Alguns sistemas armazenam dados digitados em buffers temporários antes de processá-los. Keyloggers que exploram essa técnica acessam esses buffers de memória para registrar teclas em trânsito. Isso permite capturar informações sem depender de hooks tradicionais, tornando a detecção mais complexa.

Captura de clipboard:

Embora não capture diretamente as teclas, keyloggers complementares monitoram o conteúdo da área de transferência sempre que o usuário copia ou cola informações. Esse método é amplamente utilizado para obter senhas, chaves criptográficas e textos sensíveis copiados manualmente, especialmente quando as aplicações protegem campos de senha contra keylogging convencional.

Keylogging por overlay (Android):

Em dispositivos móveis, especialmente no Android, alguns malwares criam uma camada falsa (overlay) sobre aplicativos legítimos. Essa camada exibe campos de entrada idênticos aos originais, mas captura tudo o que o usuário digita.

Trata-se de uma abordagem visual e extremamente eficaz em ataques de phishing móvel.

Captura pré-TLS (form grabbing):

O form grabbing ocorre antes que os dados digitados em formulários web sejam criptografados pelo protocolo TLS. Em vez de capturar teclas individualmente, esse método intercepta o conteúdo completo de campos de login, senhas e formulários no navegador, tornando-o mais preciso e eficiente do que o keylogging tradicional. É amplamente usado por trojans bancários.

Hook de eventos JavaScript:

Em ambientes de navegador, scripts maliciosos podem registrar eventos como *onkeypress*, *oninput* ou *keydown* para capturar tudo o que o usuário digita em uma página adulterada. Esse método é comum em ataques de phishing, comprometimentos de sites (como Magecart) e extensões maliciosas, operando totalmente no contexto da web.

4. Arquitetura de um Keylogger

A arquitetura de um keylogger moderno costuma seguir uma estrutura modular, em que cada componente cumpre uma função específica dentro do ciclo de captura, processamento, persistência e envio dos dados coletados. Embora existam keyloggers extremamente simples, compostos apenas por rotinas de captura e armazenamento local, as variantes utilizadas em ataques reais tendem a ser mais complexas, incorporando mecanismos de persistência, canais de exfiltração flexíveis e técnicas de evasão para dificultar a detecção. Dessa forma, o keylogger funciona como um conjunto coordenado de módulos que operam de maneira silenciosa e integrada.

O primeiro elemento essencial é o **módulo de captura**, responsável por interceptar a digitação. Esse módulo pode atuar em diversos níveis do sistema — desde APIs de alto nível até drivers de kernel — dependendo do tipo de keylogger e do nível de acesso obtido pelo atacante. Seu papel é registrar eventos de teclado, sejam eles pressionamentos, liberações ou estados contínuos de tecla, convertendo-os em dados úteis. Em keyloggers mais sofisticados, essa etapa pode incluir também captura de clipboard, formulários de navegador ou dados complementares como janelas ativas e timestamps.

Em seguida, os dados capturados passam para o **módulo de armazenamento**, que normalmente funciona como um buffer temporário. Este buffer pode existir na memória, em arquivos ocultos no disco ou até em estruturas criptografadas

que aguardam o momento ideal de exfiltração. A função do módulo é organizar as informações para reduzir o volume de envio e evitar ruídos que possam chamar atenção. Alguns keyloggers implementam compressão ou criptografia nesse estágio para aumentar a furtividade.

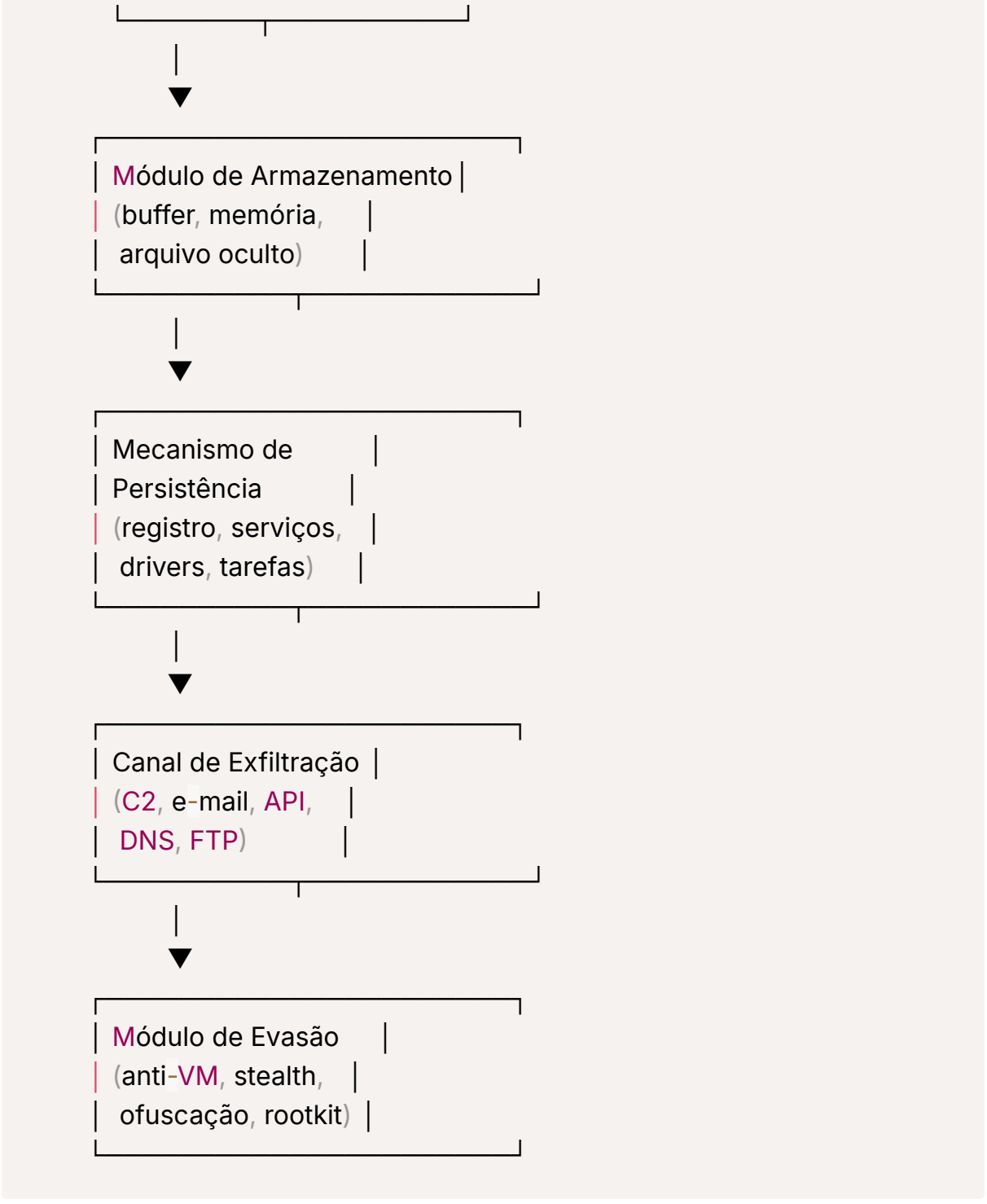
Outro componente crítico da arquitetura é o **mecanismo de persistência**, responsável por garantir que o keylogger continue operando mesmo após reinicializações do sistema ou tentativas de remoção. A persistência pode ser obtida por meio de chaves de registro, tarefas agendadas, services, carregadores no início da sessão ou, em casos mais avançados, manipulação de drivers e componentes do sistema. Quanto mais profundo o método, maior a resistência à detecção.

Com os dados já coletados e armazenados, entra em ação o **canal de exfiltração**, cuja função é enviar os logs para o atacante. Esse envio pode ocorrer de várias formas: e-mail, FTP, protocolos customizados, comunicação com um servidor C2, uso de APIs públicas (como Telegram ou Discord), ou até canais encobertos via DNS. Keyloggers modernos tentam realizar a exfiltração apenas quando o sistema está ocioso ou conectado à internet, reduzindo riscos de alerta.

Para se manter oculto, muitos keyloggers incorporam ainda um **módulo de evasão**, especializado em dificultar a ação de antivírus, EDRs e ferramentas de análise. Esse módulo pode incluir técnicas como ofuscação, detecção de ambientes virtualizados, verificação de processos de segurança, uso de rootkits ou modificação de timestamps e metadados de arquivos. A evasão é fundamental em keyloggers voltados para ataques contínuos ou campanhas direcionadas.

A combinação desses componentes cria um fluxo interno consistente: o evento de teclado é interceptado, convertido e enviado ao buffer, persistido no sistema e, eventualmente, extraído silenciosamente. Esse fluxo pode variar conforme a camada de atuação e o objetivo do ataque, mas em essência segue uma cadeia lógica de:

> captura → armazenamento → persistência → exfiltração → evasão.



5. Detecção e Prevenção

A detecção e prevenção de keyloggers é um dos pilares mais críticos em qualquer estratégia corporativa de segurança, já que esses mecanismos atacam diretamente o elo mais vulnerável do processo de autenticação: o usuário. Como sua natureza é furtiva, a mitigação eficaz depende de uma combinação equilibrada entre controles técnicos, boas práticas de navegação,

ferramentas de proteção e monitoramento contínuo. Para reduzir o risco, as empresas precisam adotar medidas preventivas sólidas, além de mecanismos de detecção capazes de identificar comportamentos anômalos, alterações no sistema e exfiltrações suspeitas.

5.1 Prevenção técnica

A primeira camada de proteção envolve o **hardening do sistema operacional**, garantindo que superfícies críticas — como permissões de pastas sensíveis, configurações de inicialização e serviços de alto privilégio — sejam rigidamente controladas. Essa etapa impede que keyloggers consigam executar rotinas em níveis elevados ou persistir via mecanismos comuns como Registro, services ou drivers. Da mesma forma, a **restrição de permissões** limita a capacidade de processos comuns de acessar APIs sensíveis, impedindo que softwares não autorizados apliquem hooks, injetem DLLs ou façam leitura de buffers internos.

Outro ponto fundamental é o **gerenciamento de extensões**, especialmente em navegadores corporativos. Extensões maliciosas representam uma das maiores fontes de keyloggers modernos, especialmente aqueles baseados em JavaScript e scripts de form grabbing. Apenas extensões corporativamente aprovadas devem ser permitidas. Além disso, práticas de **navegação segura**, com bloqueio de sites suspeitos, inspeção SSL e políticas de content filtering, reduzem a exposição a páginas adulteradas.

O **controle de dispositivos USB** também é vital, já que dispositivos físicos podem atuar como keyloggers de hardware ou emular teclados maliciosos via ataques HID. Políticas de bloqueio, listas brancas e monitoramento de inserção de dispositivos mitigam esses vetores.

Para ambientes web, especialmente e-commerce, o uso de **Content Security Policy (CSP)** e **Subresource Integrity (SRI)** é crucial. Esses mecanismos ajudam a prevenir injeções de scripts externos (skimmers) que capturam campos de formulário antes do envio. Eles aumentam significativamente a resistência contra ataques como Magecart.

No campo das proteções reativas, ferramentas como **antivírus, EDR e monitoramento comportamental** são essenciais. Essas soluções identificam atividades como keylogging por API hooking, injeção de DLL, manipulação do registro e criação de persistência. Quando aplicadas em conjunto com **updates regulares, patches e um modelo de Zero Trust**, formam um conjunto robusto

para limitar movimentação lateral e impedir que processos desconhecidos acessem áreas críticas do sistema.

5.2 Detecção

A detecção eficaz de keyloggers exige uma abordagem baseada em comportamento. A **análise de anomalias** permite identificar padrões incomuns, como processos que repentinamente começam a monitorar entradas de teclado ou a acessar APIs de captura de eventos. A **verificação de integridade de arquivos**, especialmente em diretórios sensíveis, é útil para detectar alterações ocultas, arquivos criados recentemente e componentes de persistência.

A **identificação de hooks suspeitos**, incluindo interceptações de funções de teclado, drivers carregados fora da cadeia de confiança e hooks globais aplicados por processos não autorizados, é um dos métodos mais diretos de detectar keyloggers sofisticados. O **monitoramento de processos** complementa essa análise ao observar criação de processos invisíveis, uso incomum de memória, injeções de DLL e operações de leitura contínua das APIs de entrada.

Por fim, a **análise de redes** desempenha um papel essencial na detecção de exfiltração. Keyloggers frequentemente enviam dados em períodos específicos, usando protocolos comuns (HTTP, SMTP, DNS) para mascarar o tráfego. Monitorar essas anomalias, especialmente conexões persistentes ou tráfego criptografado para destinos desconhecidos, ajuda na identificação precoce.

No contexto de e-commerce, a **inspeção de scripts** também se torna indispensável. Keyloggers baseados em skimming, como Magecart, dependem de scripts injetados no lado do cliente; auditorias periódicas do front-end, validação de integridade e análise de mudanças reduzem drasticamente esse risco.

5.3 Prevenção no contexto de CheckPoint

Dentro de ambientes que utilizam soluções Check Point, é possível reforçar significativamente a proteção contra keyloggers combinando módulos do Infinity Architecture:

1. Harmony Endpoint – Prevenção e Detecção Local:

- Previne keyloggers por API hooking usando Behavioral Guard e Anti-Bot.
- Bloqueia injeções de DLL, criação de persistência e execução de binários suspeitos.
- Monitora clipboard, drivers, hooks e comportamentos anômalos.
- Implementa Anti-Ransomware, que também detecta padrões de keylogging.

2. Threat Emulation + Threat Extraction:

- Analisa anexos e arquivos baixados que possam conter keyloggers, usando sandboxing avançado.
- Remove partes ativas de documentos (macros, scripts) evitando keyloggers embutidos.

3. Harmony Browse / HTTPs Inspection:

- Bloqueia scripts maliciosos e skimmers em páginas web.
- A aplicação de CSP e SRI pode ser reforçada com inspeção e bloqueio de conteúdo suspeito.
- Evita extensões maliciosas instaladas no navegador.

4. Anti-Bot + Threat Intelligence:

- Identifica comunicação de exfiltração com servidores C2 conhecidos.
- Bloqueia tráfego suspeito baseado em reputação, DNS, IP e URL.

5. Segmentation + Zero Trust no Check Point:

- Reduz superfícies onde o keylogger pode se propagar ou enviar dados.
- Limita o impacto de processos comprometidos através de políticas de identidade e microsegmentação.

6. SmartEvent + Logs unificados:

- Permite visualizar tentativas de exfiltração, conexões maliciosas e criação de persistência.
- Auxilia na correlação de eventos que indicam keylogging, como API hooking + comportamento de rede.

Com esses módulos combinados, um ambiente CHKP consegue impedir keyloggers na origem, mitigar sua capacidade de persistir e bloquear tentativas

de comunicação maliciosa, oferecendo múltiplas camadas de proteção.