

Análise de Segurança de um Sistema NVR Comercial por meio de Engenharia Reversa

Gabriel Haab¹, Leandro Bolzan², Érico Amaral³

¹PhD Cyber Operations – Dakota State University

²Ministerio das Comunicações - Brasil

³Engenharia de Computação – Universidade Federal do Pampa (UNIPAMPA)

Gabriel.KitzmannHaab@trojans.dsu.edu, leandroberia@unipampa.edu.br, ericoamaral@unipampa.edu.br

Abstract. *This paper describes the security analysis of a commercial NVR (Network Video Recorder) using reverse engineering techniques. The methodology involved firmware deobfuscation and analysis, file system extraction, physical access via UART (Universal Asynchronous Receiver/Transmitter), and analysis of the associated mobile application. The results identified weaknesses in credential storage and the device boot process. The main finding demonstrated the possibility of replacing the main application with an external executable and gaining root-level access to the system. The results highlight the relevance of an integrated analysis of firmware, hardware, and software for assessing the security of IoT devices.*

Resumo. *Este artigo descreve a análise da segurança de um sistema NVR (Network Video Recorder) comercial por meio de técnicas de engenharia reversa. A metodologia envolveu a desobfuscação e análise do firmware, extração do sistema de arquivos, acesso físico via UART (Universal Asynchronous Receiver/Transmitter) e análise do aplicativo móvel associado. Os resultados identificaram fragilidades no armazenamento de credenciais e no processo de inicialização do dispositivo. O principal achado demonstrou a possibilidade de substituir a aplicação principal por um executável externo e obter acesso ao sistema com privilégios de root. Os resultados evidenciam a relevância da análise integrada de firmware, hardware e software na avaliação da segurança de dispositivos IoT.*

1. Introdução

A crescente adoção de dispositivos de Internet das Coisas (Internet of Things – IoT) em ambientes residenciais, comerciais e industriais amplia as possibilidades de automação, mas também introduz novas superfícies de ataque. Mosenia e Jha (2017) destacam que ameaças à segurança podem ocorrer em diferentes níveis das arquiteturas IoT, envolvendo dispositivos, mecanismos de comunicação e serviços associados. Nesse contexto, sistemas de videomonitoramento, como os Network Video Recorders (NVRs), apresentam especial relevância por processarem informações potencialmente sensíveis e integrarem firmware, interfaces de rede, mecanismos de autenticação, serviços em nuvem e aplicativos móveis. A engenharia reversa constitui uma estratégia para avaliar esses componentes: Cui et al. (2018) demonstraram sua aplicação na investigação de dispositivos IoT, enquanto Wang et al. (2021) combinaram extração de firmware, análises estática e dinâmica e inspeção do tráfego de rede para investigar dispositivos baseados em Linux.

Neste contexto, este trabalho apresenta uma análise de segurança de um sistema comercial de videomonitoramento Firstrend NVR, equipado com oito canais sem fio, resolução 1080p, plataforma HiSilicon e firmware baseado em Linux, além de integração com

o aplicativo móvel ESeeCloud. O objetivo é avaliar a segurança de um NVR comercial de baixo custo por meio de técnicas de engenharia reversa, considerando diferentes componentes do seu ecossistema. Para isso, foram realizadas a extração e análise estática e dinâmica do firmware, o acesso ao equipamento por meio da interface UART, a investigação de scripts e mecanismos de autenticação e a análise do aplicativo Android associado.

Os resultados demonstraram a possibilidade de reverter a ofuscação XOR do firmware, acessar seu sistema de arquivos, identificar o emprego de MD5crypt para armazenamento de credenciais e recuperar uma das senhas por ataque de dicionário. O principal achado foi identificado no script `run_app.sh`, que permite substituir temporariamente a aplicação principal `app.out` por um executável fornecido externamente por uma unidade USB; a validação experimental desse comportamento possibilitou obter um shell com privilégios de root. A investigação também identificou potenciais superfícies de ataque relacionadas à autenticação em nuvem e ao aplicativo móvel, ainda não validadas experimentalmente. Todos os experimentos foram realizados em equipamento próprio e em uma rede isolada, sem exploração da infraestrutura de produção do fabricante.

O restante deste artigo está organizado da seguinte forma: a Seção 2 apresenta a fundamentação e os trabalhos relacionados; a Seção 3 descreve a metodologia utilizada para análise do dispositivo; a Seção 4 apresenta e discute os resultados obtidos; e a Seção 5 apresenta as conclusões e as perspectivas de trabalhos futuros.

2. Fundamentação e Trabalhos Relacionados

Dispositivos de Internet das Coisas (IoT) são frequentemente construídos a partir da integração entre hardware embarcado, firmware, interfaces de comunicação e serviços externos. Em muitos desses equipamentos, o firmware é baseado em Linux e incorpora componentes proprietários responsáveis pela inicialização, configuração do hardware, autenticação e comunicação com serviços remotos. A análise de segurança desses sistemas pode, portanto, exigir técnicas que ultrapassem a inspeção convencional dos serviços disponíveis na rede e permitam compreender o comportamento interno do dispositivo.

2.1 Engenharia Reversa e Análise de Firmware em IoT

A engenharia reversa constitui uma abordagem importante para a avaliação de dispositivos IoT quando o código-fonte e a documentação de seus componentes internos não estão disponíveis. A partir da obtenção do firmware, técnicas de análise estática permitem examinar sistemas de arquivos, scripts de inicialização, bibliotecas, executáveis, credenciais e configurações. A análise dinâmica, por sua vez, possibilita observar o comportamento do dispositivo durante sua execução e validar aspectos identificados previamente.

Shwartz et al. (2018) investigaram 16 dispositivos IoT comerciais utilizando técnicas de engenharia reversa de baixo custo. Os autores demonstraram que essas abordagens podem permitir a recuperação de firmware e credenciais e revelar falhas de projeto não identificadas por análises exclusivamente externas. O estudo também evidencia que mecanismos de proteção de acesso nem sempre representam uma barreira suficiente para a análise interna do dispositivo.

Liu et al. (2021) propuseram uma metodologia para engenharia reversa de protocolos de comunicação utilizados por dispositivos IoT baseados em Linux. A abordagem combina obtenção do firmware e do aplicativo associado ao dispositivo, análise do tráfego de rede e análises estática e dinâmica. Como estudo de caso, os autores analisaram um dispositivo WeMo e demonstraram que a compreensão conjunta do firmware e das comunicações

permitiu identificar problemas relacionados à autenticação e ao controle do equipamento. Essa abordagem é particularmente relevante para sistemas cujo funcionamento depende da interação entre dispositivo físico, aplicação móvel e infraestrutura de rede.

No equipamento investigado neste trabalho, essas características também estão presentes. O NVR analisado utiliza firmware Linux sobre uma plataforma HiSilicon e incorpora módulos proprietários para funções de processamento de vídeo e comunicação. A ausência de documentação desses componentes tornou necessária a combinação entre análise estática e execução no dispositivo físico. A tentativa inicial de emulação por meio do QEMU (Quick Emulator) não foi suficiente devido à ausência de suporte aos periféricos proprietários da plataforma, levando à adoção de acesso físico via UART para a análise dinâmica.

2.2 Segurança em Sistemas de Videomonitoramento

Sistemas de videomonitoramento conectados apresentam características que tornam sua segurança particularmente relevante. Além de permanecerem continuamente conectados à rede, esses equipamentos processam imagens de ambientes residenciais ou comerciais e frequentemente oferecem acesso remoto por meio de aplicativos móveis e serviços em nuvem. Consequentemente, uma fragilidade pode afetar não apenas a disponibilidade do dispositivo, mas também a confidencialidade das informações processadas.

Stabili et al. (2024) realizaram uma análise de segurança da câmera IP Tenda CP3 utilizando técnicas estáticas e dinâmicas sobre executáveis extraídos do firmware. Diferentemente de avaliações limitadas à observação do tráfego de rede, os autores investigaram o comportamento interno do dispositivo e identificaram cinco vulnerabilidades posteriormente associadas a CVEs, demonstrando a importância da análise de firmware para a identificação de vulnerabilidades em equipamentos comerciais de videomonitoramento. O presente trabalho aproxima-se dessas pesquisas ao utilizar engenharia reversa e análises estática e dinâmica para investigar um dispositivo IoT comercial. Entretanto, diferencia-se pelo objeto e pela abrangência da análise: em vez de uma câmera IP individual, investiga-se um NVR de baixo custo baseado em HiSilicon, considerando diferentes componentes do seu ecossistema. A análise compreende a proteção e extração do firmware, o processo de inicialização, o acesso físico por UART, os mecanismos de autenticação, a aplicação principal do equipamento e o aplicativo Android utilizado para acesso remoto. No dispositivo analisado, a investigação do firmware permitiu identificar, entre outros aspectos, um mecanismo no script `run_app.sh` capaz de substituir a aplicação principal por um executável fornecido externamente, posteriormente validado experimentalmente no equipamento.

Dessa forma, este estudo contribui para a literatura ao apresentar uma análise experimental de segurança em múltiplas camadas de um sistema NVR comercial, relacionando aspectos de firmware, hardware, autenticação e aplicação móvel. Essa perspectiva permite avaliar não apenas vulnerabilidades isoladas, mas também como decisões de implementação em diferentes componentes podem ampliar a superfície de ataque de sistemas IoT de videomonitoramento.

3. Metodologia

A metodologia consistiu na engenharia reversa e análise de segurança de um Firstrend NVR baseado em plataforma HiSilicon e firmware Linux. O processo envolveu a obtenção e desofuscação XOR do firmware, extração do sistema de arquivos JFFS2 (Journalling Flash File System) e análise estática do script `run_app.sh`, do executável `app.out` e das credenciais armazenadas, utilizando o Ghidra. Como a emulação via QEMU mostrou-se inviável devido às dependências de hardware proprietárias, a análise dinâmica foi realizada diretamente no

equipamento por meio de uma interface UART, permitindo acompanhar sua inicialização e interagir com o sistema.

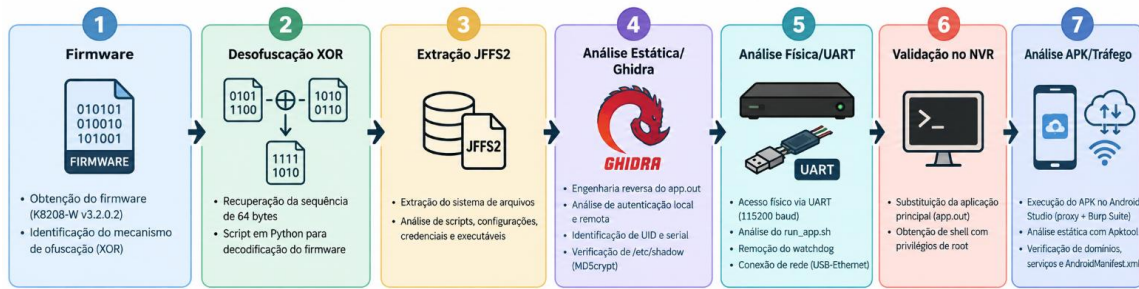


Figura 1. Visão geral da metodologia

A Figura 1 sintetiza o fluxo metodológico adotado no estudo, abrangendo desde a obtenção e desofuscação do firmware até as análises estática e dinâmica do NVR e a avaliação do aplicativo móvel e de seu tráfego de rede. Complementarmente, foi analisado o aplicativo Android ESeeCloud, utilizado para acesso remoto ao NVR. O APK (Android Package Kit) foi executado em um emulador do Android Studio e seu tráfego foi direcionado ao Burp Suite, permitindo observar a comunicação entre a aplicação e os serviços externos. Também foi realizada análise estática após a descompilação com Apktool, examinando domínios, serviços e configurações presentes no AndroidManifest.xml. Todos os experimentos foram conduzidos em equipamento próprio e em rede isolada, sem exploração da infraestrutura de produção do fabricante. Os achados não validados experimentalmente foram mantidos separados dos resultados comprovados durante os testes.

4. Resultados e Discussões

A aplicação da metodologia permitiu identificar fragilidades em diferentes componentes do NVR, desde a proteção do firmware até seu processo de inicialização e mecanismos de autenticação. Para evitar atribuir o mesmo nível de evidência a todos os resultados, os achados foram classificados entre aqueles validados experimentalmente e aqueles que representam potenciais superfícies de ataque, identificadas durante a engenharia reversa, mas ainda não exploradas. Os principais resultados são apresentados a seguir.

4.1 Desofuscação do Firmware e Análise de Credenciais

A análise da imagem do firmware revelou a utilização de um mecanismo de ofuscação baseado em XOR com uma sequência de 64 bytes. A recuperação dessa sequência e sua aplicação por meio do script desenvolvido durante o estudo possibilitaram decodificar o firmware e acessar seu sistema de arquivos. Esse resultado demonstra que o mecanismo utilizado representa uma barreira à inspeção direta, mas não fornece proteção suficiente contra a recuperação do conteúdo quando seu funcionamento é identificado.

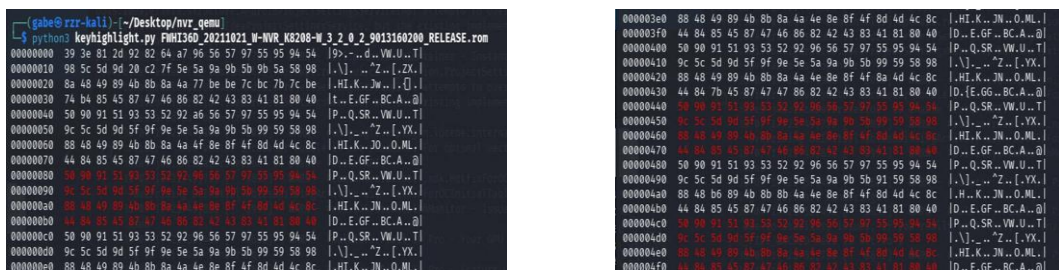


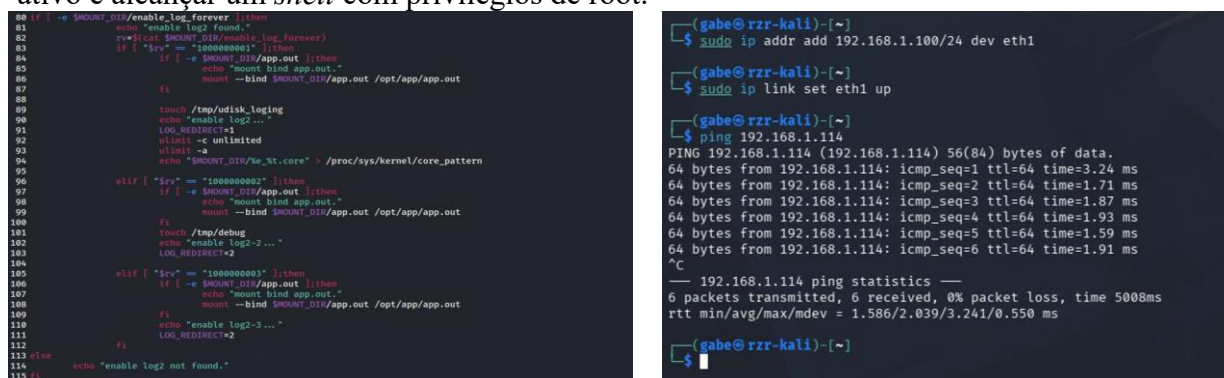
Figura 2 – Identificação da sequência utilizada na ofuscação XOR do firmware.

A inspeção do arquivo `/etc/shadow` identificou ainda a utilização de MD5crypt para armazenamento das credenciais. Os hashes recuperados foram submetidos a ataques de dicionário, sendo possível descobrir a senha da conta `stb`, cuja credencial correspondia ao próprio nome da conta. A senha da conta `root`, por outro lado, não foi recuperada durante os experimentos, mesmo após testes envolvendo aproximadamente 1,1 bilhão de candidatos. Esses resultados indicam que, além da utilização de um mecanismo legado para armazenamento das senhas, a adoção de credenciais fracas pode facilitar o comprometimento das contas existentes no dispositivo.

4.2 Alteração do Processo de Inicialização e Acesso Privilegiado

O principal achado do estudo foi identificado no script de inicialização `run_app.sh`. A análise revelou que, mediante a presença de arquivos específicos em uma unidade USB, o script realiza um `bind mount` que sobrepõe a aplicação principal `app.out` por um executável fornecido externamente. Para ativar esse comportamento, o arquivo `enable_log_forever` deve conter um dos valores previstos pelo próprio script e estar acompanhado de um arquivo denominado `app.out`. Nessas condições, o executável externo passa a ser chamado no lugar da aplicação original durante a inicialização.

Esse comportamento foi reproduzido experimentalmente no NVR. A substituição controlada da aplicação possibilitou alterar o fluxo de inicialização e expor um *shell* sem autenticação pelo console serial. Durante o experimento, observou-se ainda que o módulo de *watchdog* (`wdt` - Temporizador de supervisão) reiniciava o equipamento aproximadamente a cada 30 segundos. Sua remoção durante a sessão experimental possibilitou manter o acesso ativo e alcançar um *shell* com privilégios de `root`.



The image contains two terminal screenshots. The left screenshot shows a script execution with line numbers 80 to 115. The script checks for the presence of `enable_log_forever` and `app.out`, and performs a `bind mount` if both are present. The right screenshot shows a series of network commands being executed on a Kali Linux machine, including setting the IP address to 192.168.1.100, bringing the interface up, and performing a ping to 192.168.1.114.

```

80 if [ -e $MOUNT_DIR/enable_log_forever ];then
81     echo "enable log2 found."
82     r=$(cat $MOUNT_DIR/enable_log_forever)
83     if [ "$r" = "1000000001" ];then
84         if [ -e $MOUNT_DIR/app.out ];then
85             echo "mount bind app.out"
86             mount --bind $MOUNT_DIR/app.out /opt/app/app.out
87         fi
88     fi
89     touch /tmp/udisk_logging
90     echo "enable log2 ..."
91     LOG_REDIRECT=1
92     ulimit -c unlimited
93     ulimit -a
94     echo "$MOUNT_DIR/wd_st_core" > /proc/sys/kernel/core_pattern
95
96     elif [ "$r" = "1000000002" ];then
97         if [ -e $MOUNT_DIR/app.out ];then
98             echo "mount bind app.out"
99             mount --bind $MOUNT_DIR/app.out /opt/app/app.out
100         fi
101         touch /tmp/debug
102         echo "enable log2-2 ..."
103         LOG_REDIRECT=2
104     fi
105     elif [ "$r" = "1000000003" ];then
106         if [ -e $MOUNT_DIR/app.out ];then
107             echo "mount bind app.out-2"
108             mount --bind $MOUNT_DIR/app.out /opt/app/app.out
109         fi
110         echo "enable log2-3 ..."
111         LOG_REDIRECT=2
112     fi
113 else
114     echo "enable log2 not found."
115 fi
  
```

```

(gabe@rzzr-kali)-[~]
$ sudo ip addr add 192.168.1.100/24 dev eth1
(gabe@rzzr-kali)-[~]
$ sudo ip link set eth1 up
(gabe@rzzr-kali)-[~]
$ ping 192.168.1.114
PING 192.168.1.114 (192.168.1.114): 56(84) bytes of data:
64 bytes from 192.168.1.114: icmp_seq=1 ttl=64 time=3.24 ms
64 bytes from 192.168.1.114: icmp_seq=2 ttl=64 time=1.71 ms
64 bytes from 192.168.1.114: icmp_seq=3 ttl=64 time=1.87 ms
64 bytes from 192.168.1.114: icmp_seq=4 ttl=64 time=1.93 ms
64 bytes from 192.168.1.114: icmp_seq=5 ttl=64 time=1.59 ms
64 bytes from 192.168.1.114: icmp_seq=6 ttl=64 time=1.91 ms
^C
  
```

Figura 3 – Validação do mecanismo de substituição da aplicação e obtenção de acesso privilegiado ao NVR.

Esse resultado é particularmente relevante porque demonstra como uma funcionalidade aparentemente destinada à manutenção ou depuração pode constituir um caminho para execução de código não autorizado quando não existe verificação adequada da integridade ou procedência do executável. Entretanto, o cenário validado neste estudo pressupõe acesso físico ao NVR e à interface USB, condição que deve ser considerada no modelo de ameaça e na avaliação do impacto da vulnerabilidade.

4.3 Autenticação Remota e Aplicativo ESeeCloud

A engenharia reversa do executável `app.out` permitiu identificar funções relacionadas à autenticação local e ao subsistema de comunicação P2P. O fluxo analisado compreende a obtenção do identificador do dispositivo, a leitura do UID de hardware e a geração do número serial utilizado na autenticação com os serviços remotos. A identificação de rotinas tanto de leitura quanto de escrita do UID sugere a possibilidade de alteração desse identificador;

entretanto, seus efeitos sobre a autenticação em nuvem não foram experimentalmente testados. Dessa forma, esse resultado é apresentado como uma potencial superfície de ataque, e não como vulnerabilidade comprovada.

Por fim, a análise do aplicativo Android ESeeCloud permitiu interceptar, por meio do Burp Suite, o tráfego estabelecido entre o APK e os serviços externos. Embora a interceptação tenha sido realizada com sucesso, nenhuma vulnerabilidade nessa comunicação foi comprovada durante o período do estudo. A análise estática identificou ainda diferentes serviços externos e configurações potencialmente inseguras, incluindo componentes exportados e armazenamento não criptografado relacionado a registros da aplicação. Como suas consequências não foram exploradas experimentalmente, esses elementos também são considerados pontos para investigação futura.

Os resultados demonstram, portanto, a importância de analisar conjuntamente firmware, processo de inicialização, hardware e software associado. Enquanto a análise do aplicativo e da autenticação em nuvem revelou aspectos que ainda necessitam de validação, a investigação do firmware e do dispositivo físico produziu evidências concretas de fragilidades, especialmente a possibilidade de substituição da aplicação principal e obtenção de acesso privilegiado. Esse resultado reforça a contribuição da engenharia reversa para revelar comportamentos internos que dificilmente seriam identificados por uma análise restrita aos serviços de rede.

5. Conclusões e Trabalhos Futuros

Este trabalho apresentou uma análise de segurança de um sistema NVR comercial de baixo custo por meio de técnicas de engenharia reversa, contemplando firmware, hardware e aplicativo móvel. Os experimentos permitiram reverter a ofuscação XOR do firmware, acessar seu sistema de arquivos, identificar o uso de MD5crypt para armazenamento de credenciais e recuperar uma das senhas por ataque de dicionário. O principal resultado foi a identificação e validação de um mecanismo no script `run_app.sh` que possibilita substituir a aplicação principal por um executável fornecido externamente, permitindo alterar o fluxo de inicialização e obter acesso ao sistema com privilégios de root. Como trabalhos futuros, pretende-se aprofundar a investigação dos mecanismos de autenticação em nuvem e do aplicativo ESeeCloud, além de analisar dispositivos que compartilhem plataformas ou componentes semelhantes, buscando verificar se as fragilidades identificadas também estão presentes em outros sistemas de videomonitoramento.

Referências

- Mosenia, A.; Jha, N. K. A Comprehensive Study of Security of Internet-of-Things. *IEEE Transactions on Emerging Topics in Computing*, v. 5, n. 4, p. 586–602, 2017.
- Cui, A. et al. Reverse Engineering IoT Devices: Effective Techniques and Methods. *IEEE Internet of Things Journal*, v. 5, n. 6, p. 4965–4976, 2018.
- Wang et al. On Manually Reverse Engineering Communication Protocols of Linux-Based IoT Systems. *IEEE Internet of Things Journal*, v. 8, n. 8, p. 6815–6827, 2021.
- Stabili, D.; Bocchi, T.; Valgimigli, F.; Marchetti, M. *Finding (and Exploiting) Vulnerabilities on IP Cameras: The Tenda CP3 Case Study*. 2024
- SHWARTZ, O.; MATHOV, Y.; BOHADANA, M.; ELOVICI, Y.; OREN, Y. Reverse Engineering IoT Devices: Effective Techniques and Methods. *IEEE Internet of Things Journal*, v. 5, n. 6, p. 4965–4976, 2018.