Disciplina: Segurança Cibernética

DESAFIO SHIFT REGISTER (PWN2WIN 2017)

Grupo
Arthur Hugo Barros Gaia
Felipe Ivo da Silva
Nathalia Cristina Santos
Thiago Zucarelli Crestani
Wilson de Camargo Vieira

SUMÁRIO

Introdução

- **Contexto dos CTFs de segurança computacional**
- Desafios do tipo Capture the Flag são amplamente usados para ensino, prática e avaliação em cibersegurança.
- Alguns focam especificamente em **segurança de hardware**, simulando cenários reais como engenharia reversa, verificação formal e exploração física.

- **Descrição do desafio Shift Register (Pwn2Win CTF 2017)**
- O participante deve recuperar a *flag* armazenada em um ASIC.
- O chip foi projetado com células padrão da biblioteca **OSU035**.
- São fornecidos artefatos como **arquivo .gds2** (máscara de silício), **netlist.v** e **crap.txt**.
- A tarefa principal é reconstruir a lógica do circuito e encontrar a entrada que ativa o sinal **unlocked**.

Fundamentação teórica

- **ASICs, GDSII e Bibliotecas de Células**

ASICs são circuitos integrados específicos, descritos fisicamente em GDSII.
O desafio utiliza células da biblioteca **OSU035**, aberta e amplamente documentada.

- **Shift Registers**

Cadeia de flip-flops D sincronizados que deslocam bits a cada pulso de clock.
Usados para serialização de chaves, contadores e ofuscação em hardware.
No desafio, cada bit da flag corresponde a um estágio do registrador.

- **SAT/SMT e Solver Z3**

Técnicas formais para resolver expressões lógicas.
Z3 é usado para modelagem simbólica e verificação de hardware/software.
No desafio, o Z3:

Modela a chave de 320 bits;
Representa a lógica extraída do *crap.txt*;
Encontra a entrada que torna **unlocked = 1**.

Demonstração

Metodologia

Ambiente de Execução	Artefatos do Desafio	Solver em Python 3	Dockerização do Projeto
A reprodução do desafio foi realizada em: • **Host**: Debian 13 • **Virtualização**: KVM/QEMU • **Sistema convidado**: Kali Linux • **Conteinerização**: Docker Engine Esse ambiente garante isolamento, segurança e controle sobre dependências.	O repositório oficial fornece: • ***crap.txt*** — expressões booleanas do circuito; • ***netlist.v*** — netlist gerado previamente; • ***solve.py*** — solver original em Python 2; • layout *GDS2* simplificado do circuito. Para evitar problemas de compatibilidade com Python 2, optou-se por construir um solver atualizado em Python 3.	Foi criado o arquivo solve3.py, contendo: 1. leitura e parse das expressões de *crap.txt*; 2. implementação das células lógicas (*AND, NAND, OR, INV, NOR*); 3. criação de um vetor de **320 bits** representando a chave; 4. construção do sistema de equações booleanas; 5. execução do *solver Z3*; 6. conversão do resultado simbólico para string *ASCII*. Este solver permite reprodutibilidade total dentro do Docker.	Foi criado um Dockerfile contendo: FROM python:3.11-slim WORKDIR /app COPY crap.txt netlist.v solve3.py ./ RUN pip install --no-cache-dir z3-solver CMD ["python", "solve3.py"] O build da imagem é feito com: docker build -t pwn2win-shiftreg . A execução: docker run --rm pwn2win-shiftreg

Demonstração

Resultados

A execução do solver redefinido apresenta:

> sat
>
> b'CTF-BR{A_fLaG_prINTeD_inTO_pUr3-SIlicOn}'
>
> CTF-BR{A_fLaG_prINTeD_inTO_pUr3-SIlicOn}

A solução encontrada corresponde à flag oficial documentada no write-up da equipe Dragon Sector.

Discussão

O exercício demonstrou que:

1. **Métodos formais aplicados à segurança de hardware** permitem recuperar informação protegida por lógica combinatória e sequencial.
2. Um circuito físico pode ser modelado matematicamente a partir de artefatos de síntese ou representações intermediárias.
3. O Z3 é eficaz para resolver sistemas lógicos altamente complexos sem recorrer a força bruta.
4. A dockerização garante reprodutibilidade científica, adequada a ambientes acadêmicos.

Além disso, o desafio simula cenários reais de:

- análise de IPs de hardware;
- ataque a mecanismos de proteção embutidos em silício;
- auditoria de verificação formal pré-fabricação.

Conclusão

A reprodução do desafio *Shift Register* foi completada com sucesso, demonstrando a recuperação total da flag a partir da modelagem lógica e resolução simbólica. A conteinerização via Docker permitiu independência do ambiente e facilidade de compartilhamento.

O estudo reforça a importância de:

- engenharia reversa de hardware;
- verificação formal;
- segurança em níveis abaixo do software.

O grupo conclui que a integração entre hardware e métodos formais é fundamental para a formação avançada em cibersegurança.

Referência

1. Q3K. Pwn2Win 2017 – Shift Register Challenge Repository. GitHub. Disponível em: https://github.com/q3k/ctf/tree/master/Pwn2Win2017
2. Dragon Sector. Pwn2Win 2017 – Shift Register (Write-up). https://blog.dragonsector.pl/2017/10/pwn2win-2017-shift-register.html
3. Microsoft Research. Z3 Prover. https://github.com/Z3Prover/z3
4. Oklahoma State University. OSU Standard Cell Library. Documentação pública para células OSU035.
5. Weste, N.; Harris, D. CMOS VLSI Design. Addison Wesley, 2010.
6. Marques-Silva, J.; Sakallah, K. Boolean Satisfiability in Electronic Design Automation. ACM/IEEE DAC, 2000.

Muito obrigado!