RELATÓRIO TÉCNICO – REPRODUÇÃO DO DESAFIO SHIFT REGISTER (PWN2WIN 2017)

Universidade Federal de São Carlos – UFSCar

Programa de Pós-Graduação em Ciência da Computação – PPGCC

Disciplina: Segurança Cibernética

Integrantes do Grupo de Estudos

Nome	**RA**
Arthur Hugo Barros Gaia	846602
Felipe Ivo da Silva	824079
Nathalia Cristina Santos	795698
Thiago Zucarelli Crestani	850607
Welison de Camargo Vieira	850609

RESUMO

Este relatório apresenta a reprodução completa do desafio *Shift Register*, parte do CTF **Pwn2Win 2017**, em ambiente controlado utilizando Docker e ferramentas de verificação formal. O desafio consiste em um ASIC cuja lógica de desbloqueio (sinal `unlocked`) depende de uma sequência de 320 bits (40 bytes) que percorrem um registrador de deslocamento e redes combinatórias.

Para resolver o desafio, foi realizada a modelagem do circuito com base no arquivo `crap.txt`, que contém a descrição simbólica das portas lógicas, e a utilização do solver **Z3** para encontrar uma atribuição de bits que satisfaça a condição `unlocked = 1`. O ambiente foi totalmente containerizado em Docker para garantir reprodutibilidade, isolamento e portabilidade.

O resultado final confirma a flag oficial do desafio:

CTF-BR{A_fLaG_prINTeD_inTO_pUr3-SIlicOn}

O estudo demonstra a aplicação prática de verificação formal, engenharia reversa de hardware e modelagem de circuitos digitais em desafios de segurança cibernética.

---

1. INTRODUÇÃO

Competições do tipo Capture the Flag (CTF) desempenham papel fundamental no desenvolvimento de habilidades avançadas em segurança da informação e computação. Em especial, desafios orientados a hardware buscam reproduzir cenários reais de engenharia reversa, verificação lógica e extração de segredos a partir de circuitos físicos ou representações de baixo nível, como netlists e diagramas de layout.

O desafio **Shift Register**, originalmente apresentado no Pwn2Win 2017, consiste em analisar um circuito integrado responsável por habilitar um mecanismo fictício de lançamento de foguetes. A flag, codificada diretamente no silício, só pode ser recuperada mediante a compreensão da lógica combinatória e sequencial implementada na estrutura do chip.

O presente relatório registra todas as etapas de reprodução do desafio em ambiente seguro, documentando a preparação do ambiente Docker, reconstrução do solver em Python 3 e interpretação dos resultados.

---

2. FUNDAMENTAÇÃO TEÓRICA

2.1 ASICs e Arquivos GDSII

ASICs (*Application-Specific Integrated Circuits*) são circuitos integrados projetados para funções específicas. O formato **GDSII** é amplamente utilizado na indústria microeletrônica para representar a geometria de layout das camadas do chip. Embora GDSII não contenha diretamente informações lógicas, é possível reconstruir conectividade a partir de análise geométrica, como feito no write-up original do desafio.

2.2 Células Lógicas e a Biblioteca OSU035

A biblioteca aberta **OSU035** (Oklahoma State University) fornece células padrão como AND, OR, NAND, NOR, INV e flip-flops D, que servem como blocos fundamentais para síntese digital. O desafio utiliza instâncias dessas células, sendo essencial compreender sua funcionalidade na modelagem lógica.

2.3 Registradores de Deslocamento (Shift Registers)

Um **shift register** é composto por uma cadeia de flip-flops sincronizados. Cada ciclo de clock desloca o conteúdo internamente, permitindo a entrada serial de dados e a estabilização de estados internos. No desafio, cada bit da chave secreta ocupa um estágio dessa cadeia.

2.4 Resolução Simbólica e SMT Solvers

Solvers SMT (*Satisfiability Modulo Theories*), como o **Z3**, são capazes de determinar atribuições para variáveis que satisfazem expressões lógicas. Este método é amplamente utilizado em:

- verificação formal de hardware;
- análise de segurança;
- prova automática de propriedades;
- model checking.

No contexto deste trabalho, o solver identifica a chave correta que faz `unlocked = 1`.

---

3. METODOLOGIA

3.1 Ambiente de Execução

A reprodução do desafio foi realizada em:

- **Host:** Debian 13
- **Virtualização:** KVM/QEMU
- **Sistema convidado:** Kali Linux
- **Containerização:** Docker Engine (versão atual estável)

Esse ambiente garante isolamento, segurança e controle sobre dependências.

3.2 Artefatos do Desafio

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.

3.3 Construção do Solver 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.

3.4 Dockerização do Projeto

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
```

---

4. 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.

---

5. 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.

---

6. 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 containerização 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.

REFERÊNCIAS

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.