Otisky aplikací a OS

Státnicová otázka - síťový fingerprinting, JA3, JA3S, JA4

Otisk aplikace / OS

Otisk aplikace nebo operačního systému je sada pozorovatelných vlastností komunikace, podle které lze identifikovat konkrétní software, knihovnu, verzi nebo platformu, aniž bychom museli mít přímý přístup k zařízení.

Používá se například v IDS/IPS systémech, threat intelligence, detekci malwaru, síťové forenzice, profilování klientů, detekci anomálií a pasivní identifikaci OS a aplikací.

Typicky se sledují vlastnosti na různých vrstvách síťového modelu:

- TCP/IP parametry
- TLS handshake
- HTTP hlavičky
- DNS dotazy
- pořadí a časování paketů
- chování při chybách
- podporované šifry a rozšíření

OS fingerprinting

OS fingerprinting se snaží určit operační systém podle toho, jak implementuje síťový stack.

Aktivní fingerprinting

U aktivního fingerprintingu se na cílový systém posílají speciálně vytvořené pakety a sledují se odpovědi. Typickým příkladem je Nmap OS detection.

Sleduje například:

- TTL
- TCP window size
- pořadí TCP options
- reakce na nestandardní pakety
- hodnoty IP ID
- chování při fragmentaci
- reakce na zavřené porty

Výhoda: přesnější. Nevýhoda: hlučné, může být detekováno firewallem nebo IDS.

Pasivní fingerprinting

U pasivního fingerprintingu se pouze pozoruje existující provoz. Typickým příkladem je p0f.

Sleduje například TTL, MSS, TCP window scale, TCP options, pořadí TCP options a charakteristické hodnoty TCP/IP stacku.

Výhoda: nenápadné. Nevýhoda: méně přesné, závisí na dostupném provozu.

Aplikační fingerprinting

Aplikační fingerprinting určuje konkrétní aplikaci, knihovnu nebo klienta podle způsobu komunikace.

Může jít například o webový prohlížeč, mobilní aplikaci, malware, knihovnu jako OpenSSL, BoringSSL, Go TLS nebo Python requests, konkrétní verzi klienta či automatizovaného bota.

Sledují se například HTTP hlavičky, jejich pořadí, User-Agent, TLS parametry, DNS chování, velikosti paketů, sekvence požadavků, typické endpointy a chování při přesměrování nebo chybě.

TLS fingerprinting

Velmi důležitou oblastí je TLS fingerprinting. U šifrované komunikace často nevidíme obsah, ale vidíme metadata handshake fáze.

TLS klient při spojení posílá zprávu ClientHello, která obsahuje například:

- podporované TLS verze
- seznam cipher suites
- TLS extensions
- elliptic curves / supported groups
- EC point formats
- ALPN
- SNI
- signature algorithms

Tyto hodnoty a jejich pořadí často závisí na konkrétní knihovně nebo aplikaci.

JA3

JA3 je metoda fingerprintingu TLS klientů. Vytváří otisk z polí ve zprávě TLS ClientHello.

JA3 typicky používá:

1. TLS version
2. cipher suites
3. TLS extensions
4. elliptic curves / supported groups
5. elliptic curve point formats

Z těchto hodnot se vytvoří řetězec a z něj obvykle MD5 hash.

```
TLSVersion,Ciphers,Extensions,EllipticCurves,ECPointFormats
```

K čemu je JA3 dobré

JA3 umožňuje rozpoznat klienta i tehdy, když je komunikace šifrovaná, User-Agent je podvržený, IP adresa se mění, malware používá HTTPS nebo komunikace prochází proxy.

Používá se pro detekci malwaru, identifikaci botů, threat hunting, korelaci síťových událostí a detekci neobvyklých TLS klientů.

Slabiny JA3

- nerozlišuje vždy přesně aplikaci
- různé aplikace mohou používat stejnou TLS knihovnu
- fingerprint lze napodobit
- moderní klienti používají náhodné nebo variabilní hodnoty
- JA3 ignoruje některé důležité části handshake
- u TLS 1.3 a novějších mechanismů může být méně stabilní

JA3S

Doplňkem je JA3S, což je fingerprint TLS serveru. Vychází ze zprávy ServerHello.

Používá například TLS version, vybranou cipher suite a TLS extensions.

JA3 + JA3S dohromady mohou pomoci určit kombinaci klient-server. Například stejný klient může komunikovat s různými servery, ale určitá kombinace JA3 a JA3S může být typická pro malware C2 komunikaci.

JA4

JA4 je novější rodina fingerprintů, která reaguje na omezení JA3. Cílem JA4 je být čitelnější, stabilnější, lépe použitelný pro moderní TLS, odolnější proti náhodnému pořadí hodnot a širší než jen TLS ClientHello.

JA4 se snaží nepracovat pouze s jednoduchým hashem celého seznamu, ale vytváří strukturovanější fingerprint.

Zohledňuje například transportní protokol, TLS verzi, SNI informaci, počet cipher suites, počet extensions, ALPN, tříděné hodnoty některých polí a vlastnosti ClientHello.

Zjednodušeně: JA3 je hlavně hash TLS ClientHello. JA4 je strukturovanější a modernější fingerprint, který lépe zvládá variabilitu dnešního TLS provozu.

Rodina JA4+

JA4 není jen jeden fingerprint. Často se mluví o rodině JA4+, která zahrnuje více typů otisků.

Fingerprint	**Význam**
JA4	TLS klient
JA4S	TLS server
JA4H	HTTP klient
JA4L	latence / síťová vzdálenost
JA4X	X.509 certifikáty
JA4SSH	SSH komunikace

Pro státnice obvykle stačí říct, že JA4 rozšiřuje myšlenku JA3 na modernější a širší sadu fingerprintů.

HTTP fingerprinting

U HTTP lze aplikace rozpoznávat podle User-Agentu, pořadí hlaviček, přítomnosti konkrétních hlaviček, Accept / Accept-Language / Accept-Encoding, Connection, HTTP/2 pseudo-headers, pořadí HTTP/2 settings, typických cookies a chování při redirectech.

Důležité je, že User-Agent lze snadno zfalšovat, ale kombinace hlaviček a jejich pořadí se falšuje obtížněji.

Například Chrome, Firefox, curl, wget, Python requests nebo Go HTTP client mají často rozdílné pořadí a sadu hlaviček.

SSH fingerprinting

U SSH se dá sledovat banner, podporované algoritmy, key exchange algoritmy, MAC algoritmy, kompresní algoritmy a pořadí nabízených metod.

To může pomoci určit například OpenSSH, Dropbear nebo libssh.

DNS fingerprinting

DNS chování může také prozradit aplikaci nebo malware.

Sleduje se například typ dotazů, pořadí dotazů, domény, intervaly, používání DoH / DoT, délka a entropie domén, NXDOMAIN chování a DGA domény u malwaru.

Certifikátové fingerprinty

U TLS serverů lze vytvářet otisky certifikátů podle hashe certifikátu, issuer, subject, SAN položek, délky klíče, algoritmu podpisu, platnosti a pořadí či obsahu X.509 extensions.

To se používá například při sledování phishingové infrastruktury nebo C2 serverů.

Využití v bezpečnosti

Otisky aplikací a OS se používají hlavně k detekci malwaru, identifikaci botů, rozpoznání zranitelných systémů, pasivnímu mapování sítě, korelaci incidentů, rozpoznání nástrojů útočníka, detekci podvrženého User-Agentu, tvorbě pravidel v IDS/IPS a threat huntingu.

Příklad: Zařízení tvrdí pomocí User-Agentu, že je Chrome na Windows, ale jeho TLS fingerprint odpovídá knihovně Python requests. To může indikovat automatizovaný skript nebo malware.

Obrana proti fingerprintingu

Možnosti omezení fingerprintingu zahrnují normalizaci síťového provozu, použití proxy nebo TLS terminace, sjednocení TLS konfigurace, rotaci fingerprintů, použití běžného klienta místo nestandardní knihovny, blokování aktivního skenování, firewall a IDS pravidla, omezení unikátních HTTP hlaviček a aktualizace OS a knihoven.

Úplně zabránit fingerprintingu je obtížné, protože i šifrovaná komunikace zanechává metadata.

Shrnutí ke státnici

Otisky aplikací a OS slouží k identifikaci systému nebo aplikace podle pozorovatelného chování v síti. OS fingerprinting využívá zejména vlastnosti TCP/IP stacku, například TTL, TCP window size a TCP options. Aplikační fingerprinting sleduje protokolové chování aplikací, například HTTP hlavičky, TLS handshake, DNS dotazy nebo SSH algoritmy.

Důležitým příkladem je JA3, což je fingerprint TLS klienta založený na hodnotách ze zprávy ClientHello, jako jsou TLS verze, cipher suites, extensions a elliptic curves. Doplňkem je JA3S pro server. Novější přístup JA4 rozšiřuje a zpřesňuje TLS fingerprinting, je čitelnější, stabilnější a patří do širší rodiny JA4+, která zahrnuje také HTTP, SSH, certifikáty a další typy otisků.

Tyto techniky jsou užitečné při detekci malwaru, botů, anomálií a při síťové forenzice, ale nejsou stoprocentní, protože fingerprinty lze někdy napodobit nebo změnit.