



## TitaniumCloud™ file reputation services

provide the industry's most comprehensive source for threat intelligence and reputation data on files.

TitaniumCloud identifies files and provides rich information about their contents. Every sample is processed using the ReversingLabs TitaniumCore™ Automated Static Analysis Decomposition engine to extract all contained objects and their internal Proactive Threat Indicators (PTIs). The samples are recursively unpacked, decompressed, decrypted and de-obfuscated. The proactive indicators extracted from the resulting components includes format, format validation, strings, sections and certificate chains. Malware samples are also continuously re-scanned providing historical detection information and their detection history is stored in the TitaniumCloud database.

Powerful query functions are accessed through a high performance REST Interface. **TitaniumCloud** uses a proprietary NoSQL database optimized to support advanced search and record updates across billions of file records in milliseconds.

## FEATURES

### Threat intelligence and reputation data on files

- XREF – 800+ million malware samples
- GREF – 1.7+ billion known goodware samples

### Over 3000 threat indicators per file

- Indicators include file reputation, anti-virus detection history and detailed internal information
- File source information and trust factor for all goodware samples
- Historic anti-virus scan data and threat factors
- Malware repository re-scanned continuously for changes to detection information

### Query services

- Search for files by hash or anti-virus detection name
- Bulk queries supports multi-file searches

### Performance scales to meet stringent requirements

- Bulk queries supports multi-file searches

### Performance scales to meet stringent requirements

- 8+ billion online queries per day capacity
- Optional on-premises appliance with database instance for low latency access and privacy

### Sample downloads access for qualified customers

# USE CASES

## Forensics and High Volume Classification

- TitaniumCloud reduces wasted analysis time and assets by identifying new samples as known malware or goodware through an automated REST Web Services interface.

## Incident Response and Triage

- TitaniumCloud makes analysts more productive by quickly identifying and providing information on known goodware and malware.
- Analysts can search for samples with the same anti-virus detection malware name for comparison to learn more about a threat.
- Analysts can search for similar samples using threat indicators, e.g., same section hash, embedded domain name.

## RHA1

- Analysts can uncover similar hashes leveraging ReversingLabs Similarity Hashing Algorithm (RHA1)

## Software Validation

- TitaniumCloud can identify malware and unknown components within an installation package.
- Analysts can proactively learn about false positive anti-virus detections on their company's software through alerts from TitaniumCloud.

## OEM Solution

- Software and appliance vendors can integrate TitaniumCloud into their products to quickly identify known software.
- TitaniumCloud provides a REST interface that scales for high performance requirements.

# TITANIUMCLOUD PLATFORMS

## A1000 Malware Analysis Platform

The A1000, available as an online service or on-premises appliance, enables users to search TitaniumCloud for file information using a web GUI.

## REST Web Services Over the Internet

Customer applications access TitaniumCloud REST Web Services API over the Internet. Results are returned in JSON or XML format.

## On-premises T1000 Appliance

The T1000 Appliance provides high performance, low latency solution for high volume, automated applications. The appliance maintains a local copy of the TitaniumCloud File Reputation Service on a customer's premises which is updated in real-time over the Internet.

## On-premises AT1000 Portable Incident Response Lab

The AT1000 plug and play appliance integrates with industry leading forensics solutions to significantly increase the speed and effectiveness of cyber investigations.

# DATA SET OPTIONS

## GEF

Access to information on over 1.4 billion known non-malicious files

## XREF

Access to information on over 600 million known malicious files

## Sample Access

Access to all files for qualified customers

```
XML
<...>
<status>MALICIOUS</status>- malware presence status designation
[UNKNOWN, KNOWN, SUSPICIOUS, or MALICIOUS]
<query_hash>- hash value used to execute the query; can be md5, sha1, or sha256
<sha1>1d412db0ac58dd8dbfae8b1e07b355bd14dab2c</sha1>
</query_hash>
<scanner_count>28</scanner_count>- number of scanners used in the last scan
<scanner_match>24</scanner_match>- number of scanners that detected malware in the last scan
<scanner_percent>85.7142857143</scanner_percent>- percent of scanners that detected
malware in the last scan
<first_seen>2012-07-12T00:05:00</first_seen>- first scan for the queried sample
<last_seen>2013-04-23T14:39:00</last_seen>- last scan for the queried sample
<threat_name>Win32.Trojan.Agent</threat_name>- threat name for the queried sample
<threat_level>2</threat_level>- threat level for the queried sample [0 is no threat, 1 is the smallest threat,
e.g. Adware, 5 is the biggest threat, e.g. Trojan]
<classification>- malware classification for the queried sample based on the latest scan
<family_name>Agent</family_name>- malware family name
<platform>Win32</platform>- platform
<type>Trojan</type>- malware type
<is_generic>false</is_generic>- true if malware is generically detected
</classification>
<trust_factor>5</trust_factor>- trust factor for the sample's sources
[0 is the most trusted, 5 is the least trusted]
</...>
```

```
post payload
<?xml version="1.0" encoding="UTF-8"?>
<rl>
<query>
<hash_type>md5</hash_type>
<hashes>
<item>4bb64c06b1a72539e6d3476891daf17b</item>
<item>6353de8f339b7dccc6b25356f5fbffa4e</item>
<item>59cb087c4c3d251474ded9e156964d5d</item>
<item>6c2eb9d1a094d362bcc7631f2551f5a4</item>
<item>a82c781ce0f43d06c28fe5fc8ebb1ca9</item>
<item>920f5ba4d08f251541c5419ea5fb3fb3</item>
</hashes>
</query>
</rl>
```

```
XML
<rl>- default RL root
<malware_presence>- parent node for malware presence data
<status>KNOWN</status>- malware presence status designation (UNKNOWN, KNOWN, SUSPICIOUS, or
MALICIOUS)
<query_hash>- hash value used to execute the query; can be md5, sha1, or sha256
<sha1>cf3e764af2be3711ce1147fa762562188b57dae9</sha1>
</query_hash>
</malware_presence>
</rl>

JSON
{ "rl": { "- default RL root
malware_presence": { "- parent node for malware presence data
status": "KNOWN", "- malware presence status designation (UNKNOWN, KNOWN, or SUSPICIOUS)
query_hash": { "- hash value used to execute the query; can be md5, sha1, or sha256
sha1": "cf3e764af2be3711ce1147fa762562188b57dae9"
}}}}
```

```
JSON
... {
"status": "MALICIOUS", -malware presence status designation (UNKNOWN, KNOWN, SUSPICIOUS, or
MALICIOUS)
"query_hash": { "-hash value used to execute the query; can be md5, sha1, or sha256
"sha1": "1d412db0ac58dd8dbfae8b1e07b355bd14dab2c"
},
"scanner_count": 28, -number of scanners used in the last scan
"scanner_match": 24, -number of scanners that detected malware in the last scan
"scanner_percent": 85.7142857143, - percent of scanners that detected malware in the last scan
"first_seen": "2012-07-12T00:05:00", - first scan for the queried sample
"last_seen": "2013-04-23T14:39:00", - last scan for the queried sample
"threat_name": "Win32.Trojan.Agent", - threat name for the queried sample
"threat_level": 2, - threat level for the queried sample [0 is no threat, 1 is the smallest threat, e.g. Adware, 5 is
the biggest threat, e.g. Trojan]
"classification": { "- malware classification for the queried sample based on the latest scan
"family_name": "Agent", - malware family name
"platform": "Win32", - platform
"type": "Trojan", - malware type
"is_generic": false - true if malware is generically detected
},
"trust_factor": 5 - trust factor for the sample's sources [0 is the most trusted, 5 is the least trusted]
},
}
```

```
XML
<rl>
<query>
<hash_type>hash_type</hash_type>
<hashes>
<item>hash_value</item>
<item>hash_value</item>
...
<item>hash_value</item>
</hashes>
</query>
</rl>

JSON
{ "rl": {
"query": {
"hash_type": "hash_type",
"hashes": [
"hash_value",
"hash_value",
...
"hash_value"
]
}
}}
```

```
{ "rl": {
"sync": 1
"filter": {
"updated_xref",
"updated_analysis",
"updated_source"
},
"range": {
"incremental": {
"from": "2013-03-19T13:00:00",
"to": "2013-03-19T23:00:00"
}
}
}}
```