INE– Cyber Sec

IHRP

v1

Incident Handling & Response Professional

SMTP, DNS & HTTP(S) Analytics

Section 04 | Module 03

OUTLINE

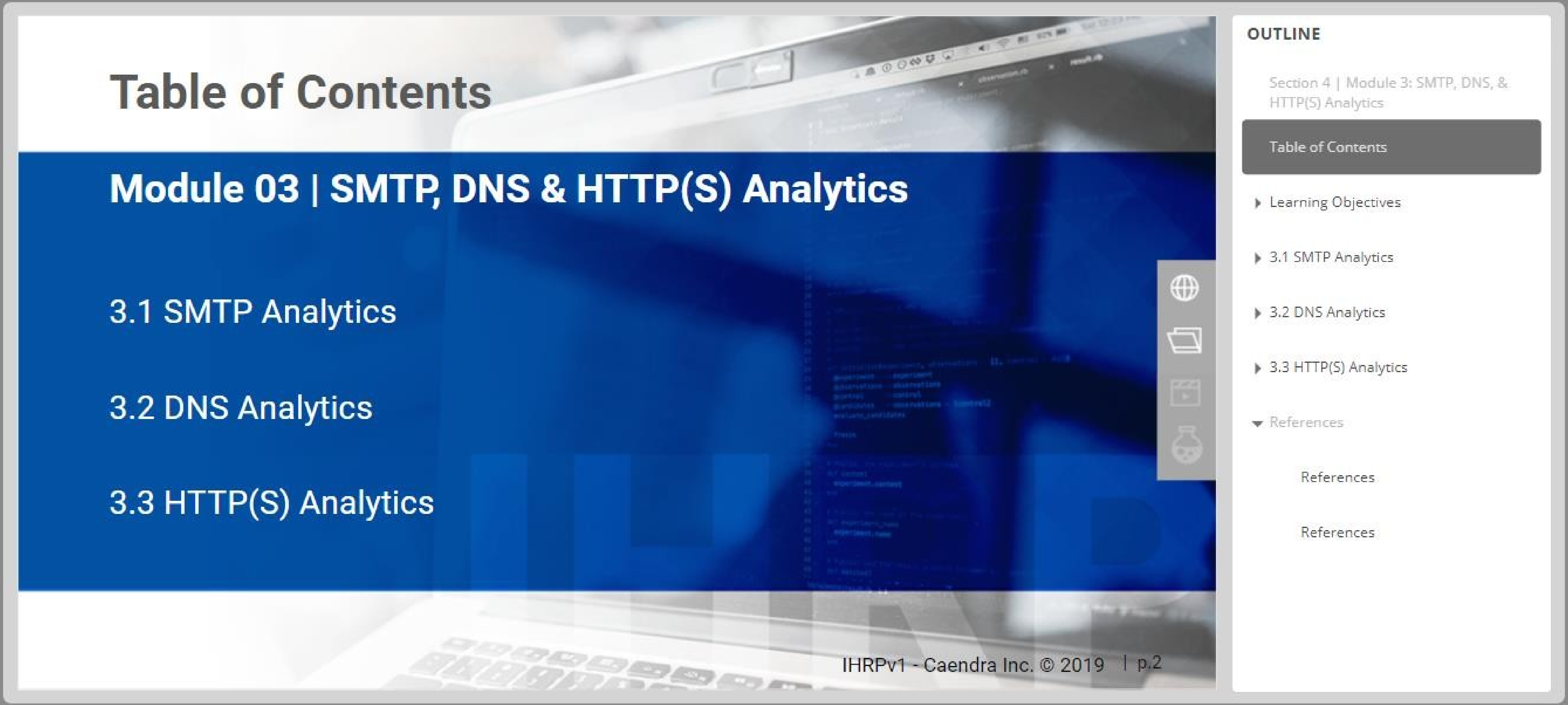
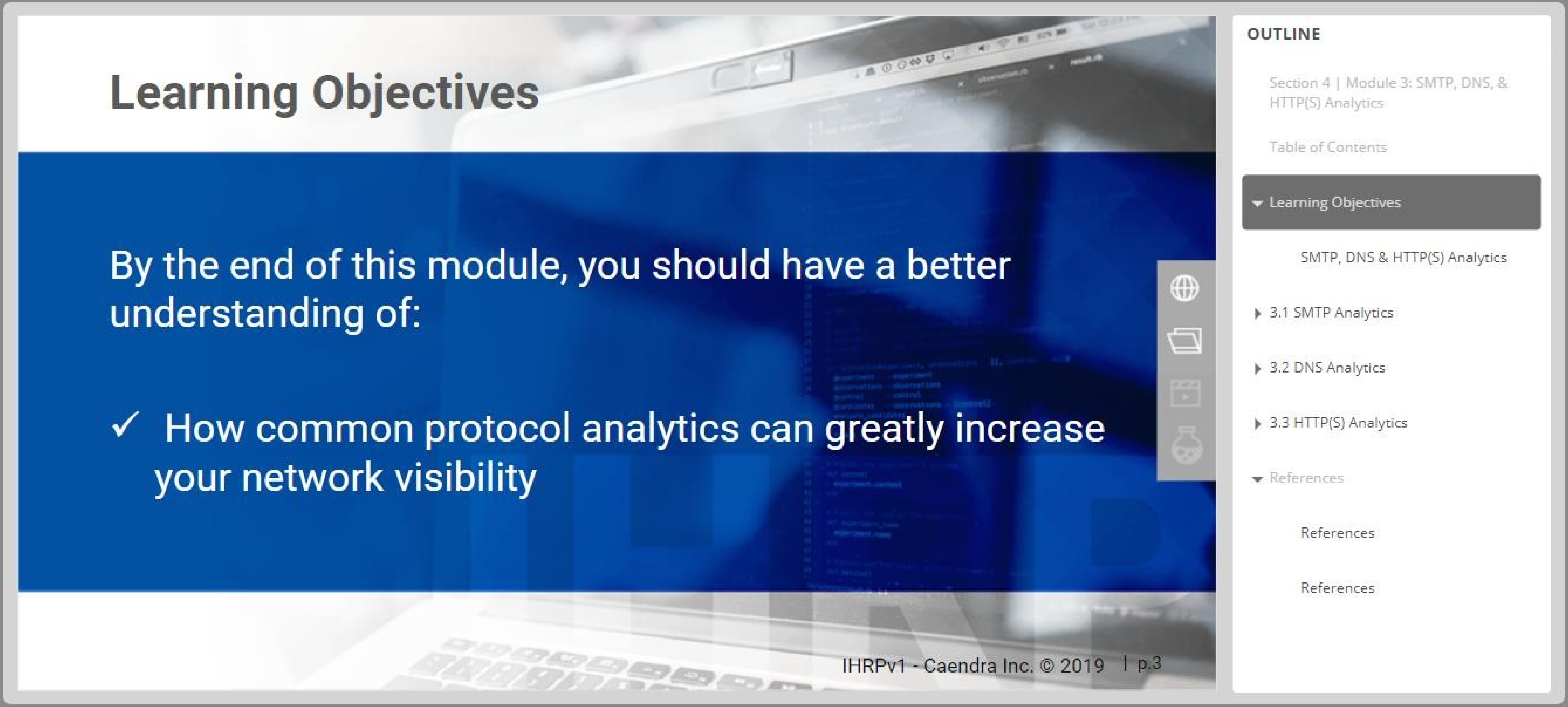


Table of Contents

Module 03 | SMTP, DNS & HTTP(S) Analytics

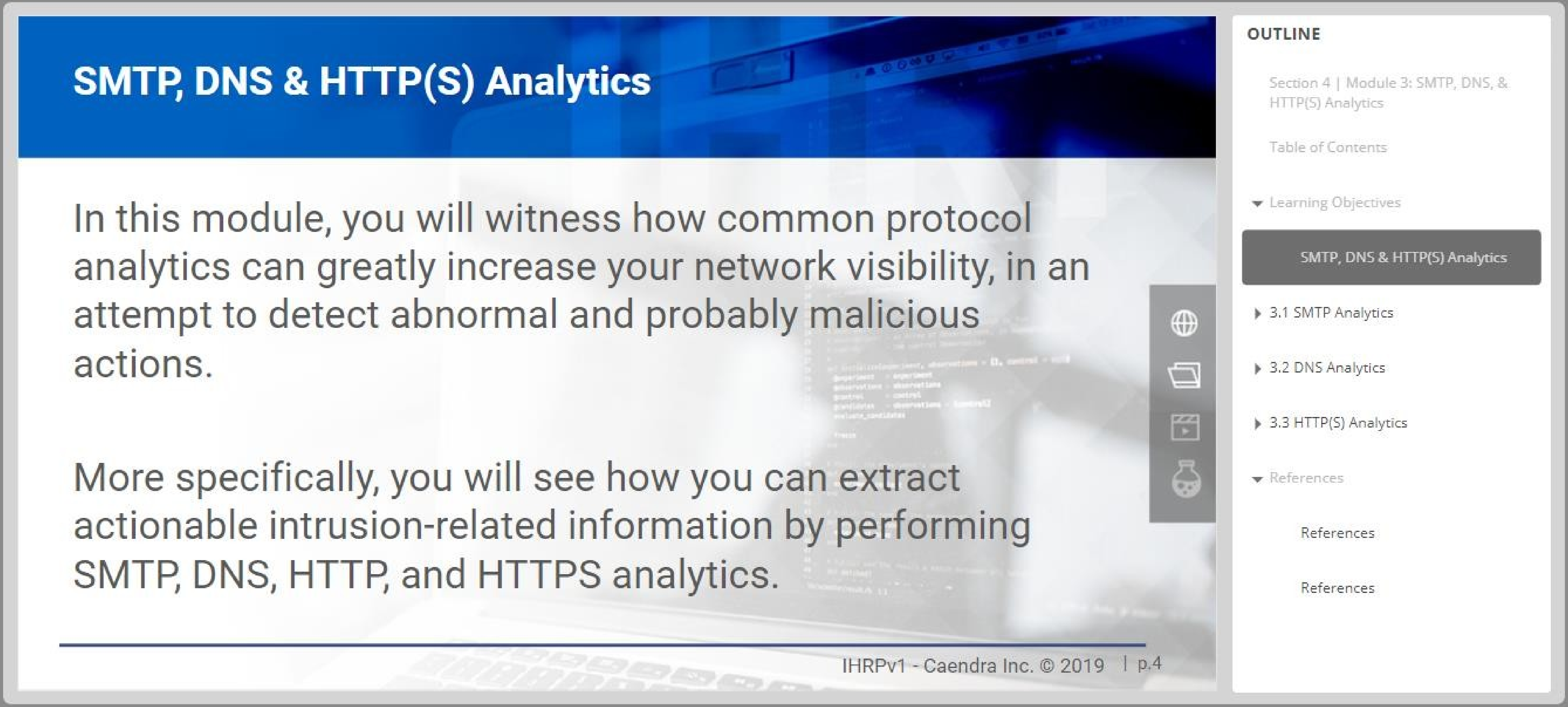


Learning Objectives

By the end of this module, you should have a better understanding of:

✓ How common protocol analytics can greatly increase your network visibility

OUTLINE



SMTP, DNS & HTTP(S) Analytics

In this module, you will witness how common protocol analytics can greatly increase your network visibility, in an attempt to detect abnormal and probably malicious actions.

More specifically, you will see how you can extract actionable intrusion-related information by performing SMTP, DNS, HTTP, and HTTPS analytics.

3.1

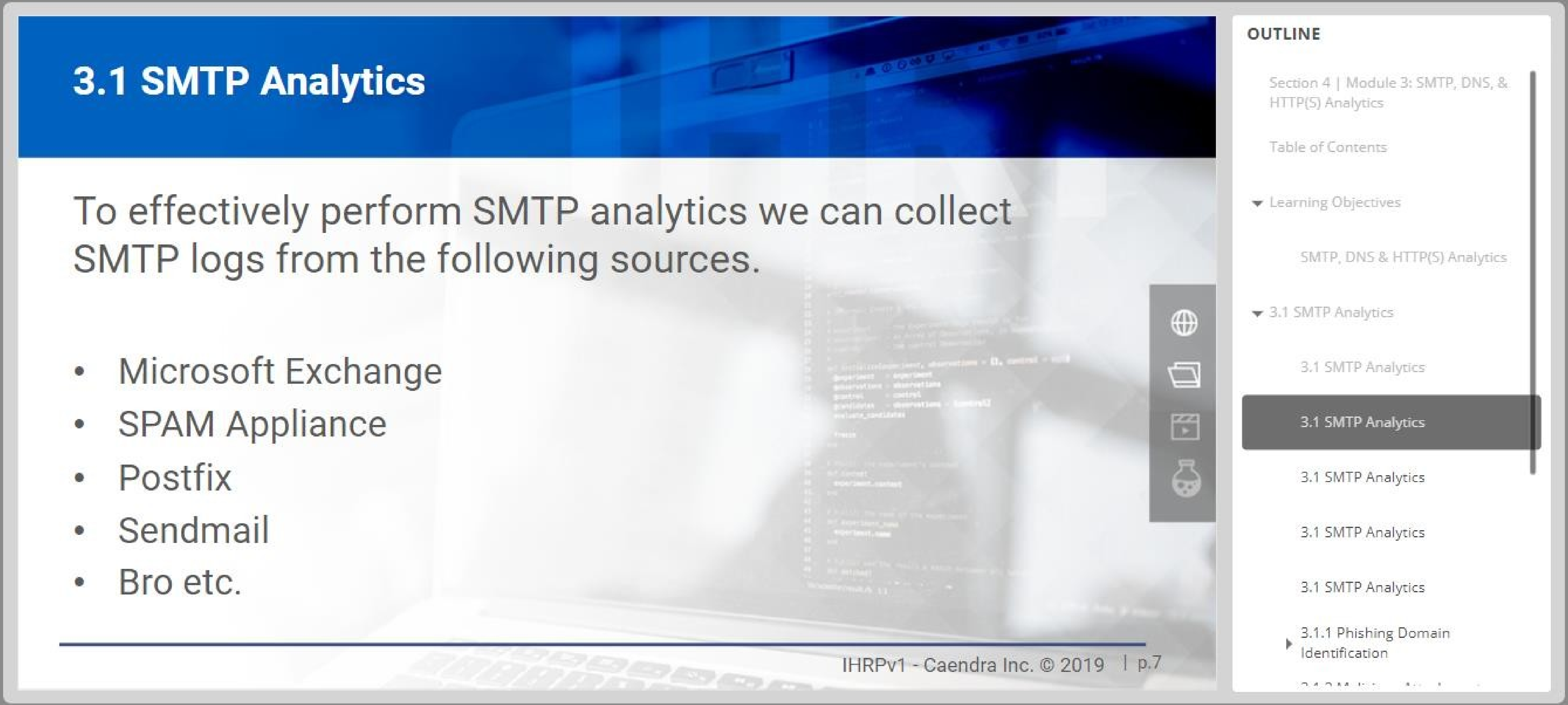
SMTP Analytics

OUTLINE

3.1 SMTP Analytics

There is no doubt that phishing remains the top threat vector for cyber attacks. Attacking the human factor continues to be the most attractive and successful path for gaining an initial foothold.

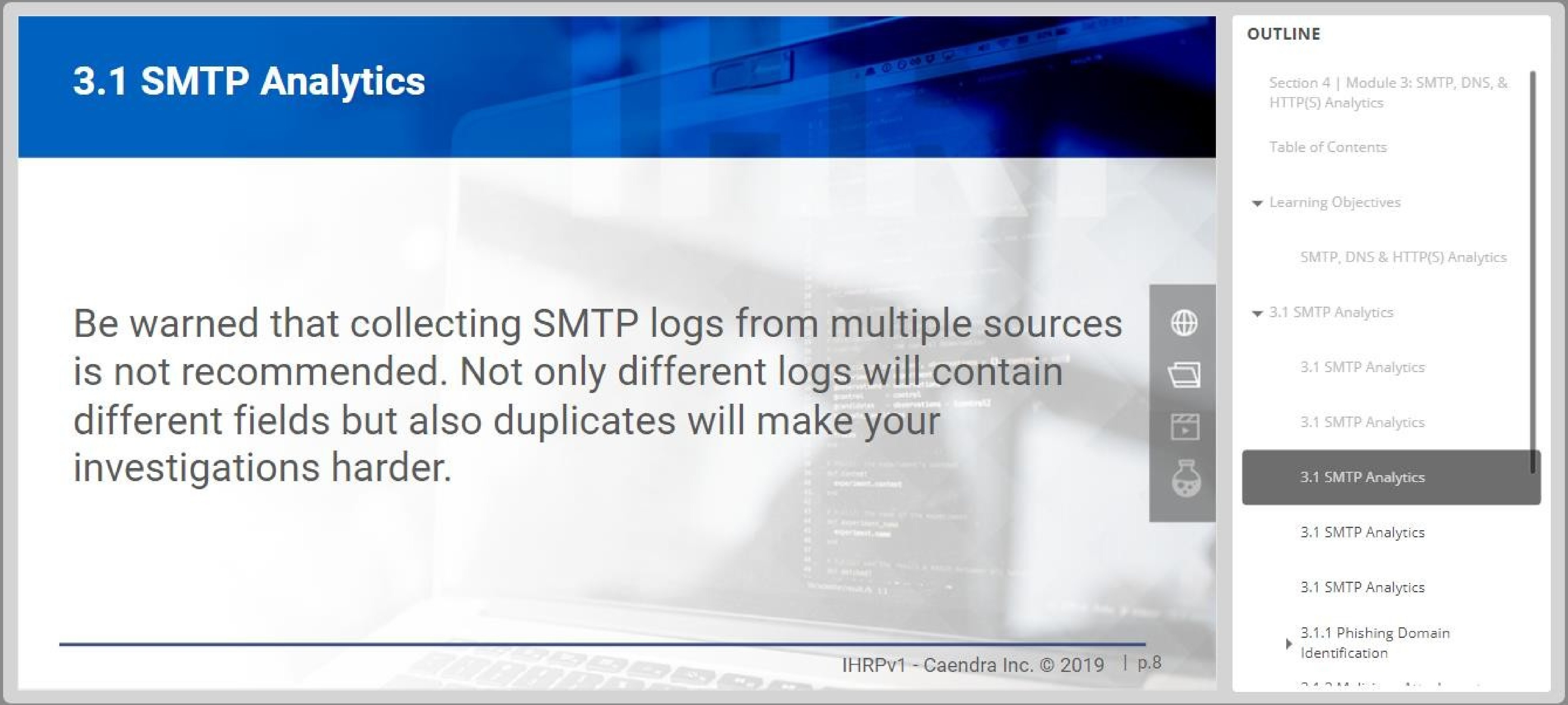
By performing SMTP analytics we can extend our visibility and detect phishing attempts.



3.1 SMTP Analytics

To effectively perform SMTP analytics we can collect SMTP logs from the following sources.

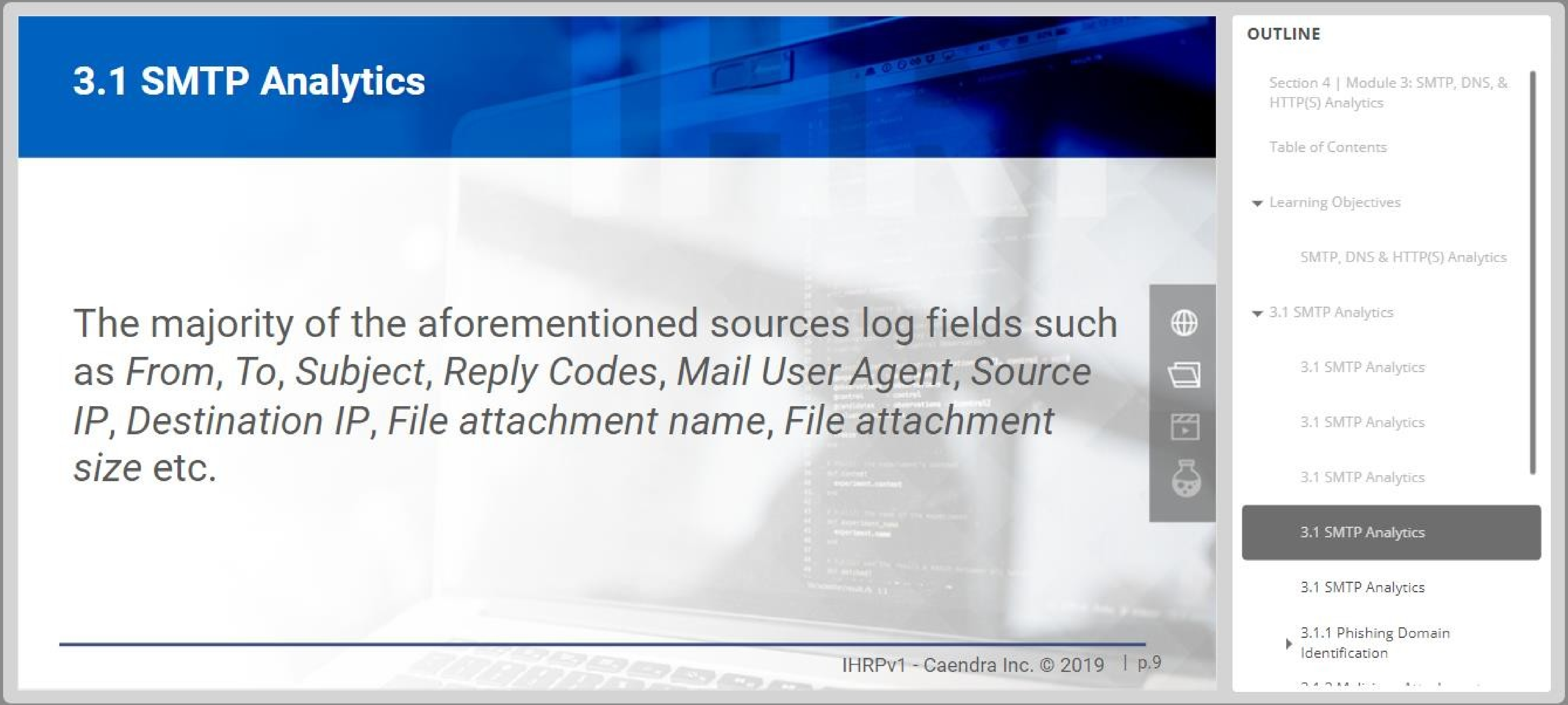
- Microsoft Exchange
- SPAM Appliance
- Postfix
- Sendmail
- Bro etc.



3.1 SMTP Analytics

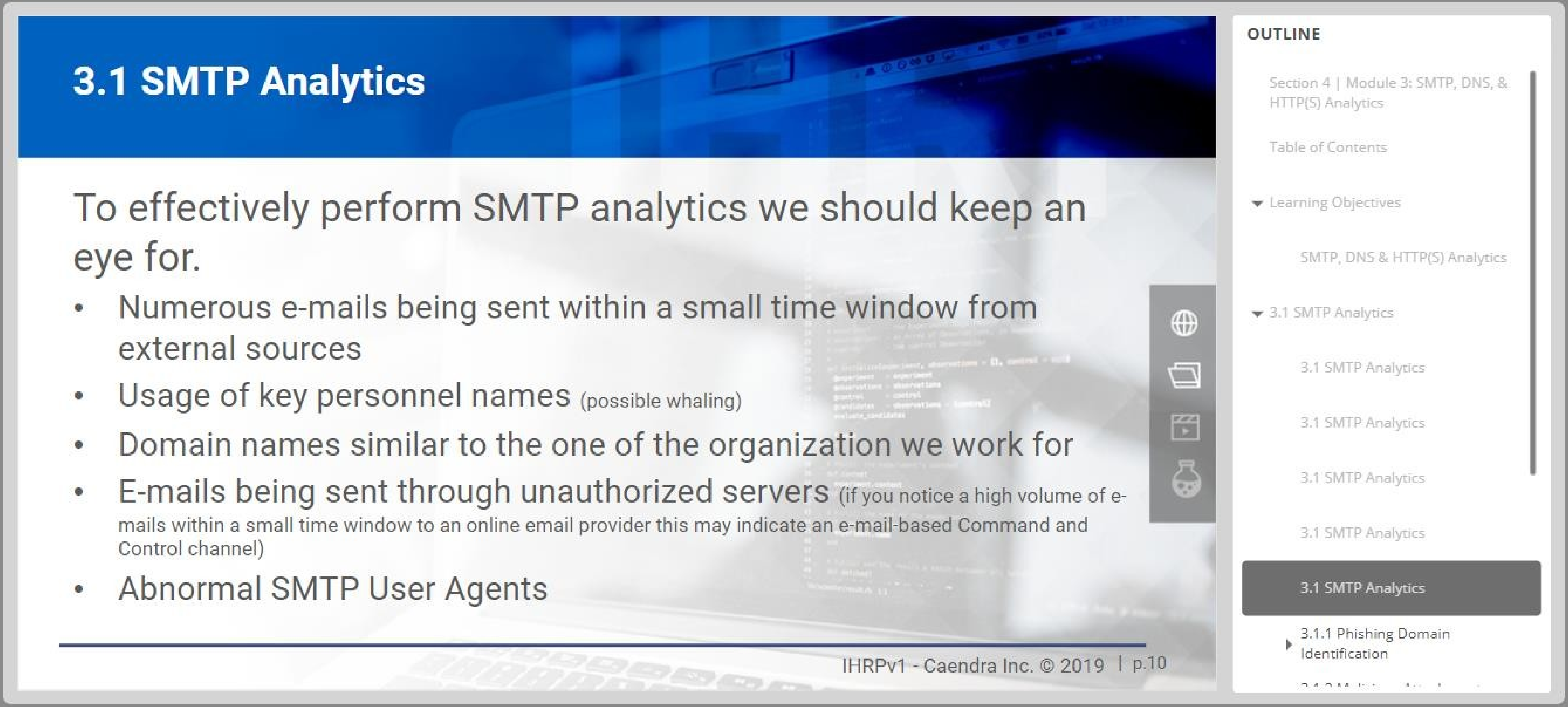
Be warned that collecting SMTP logs from multiple sources is not recommended. Not only different logs will contain different fields but also duplicates will make your investigations harder.

OUTLINE



3.1 SMTP Analytics

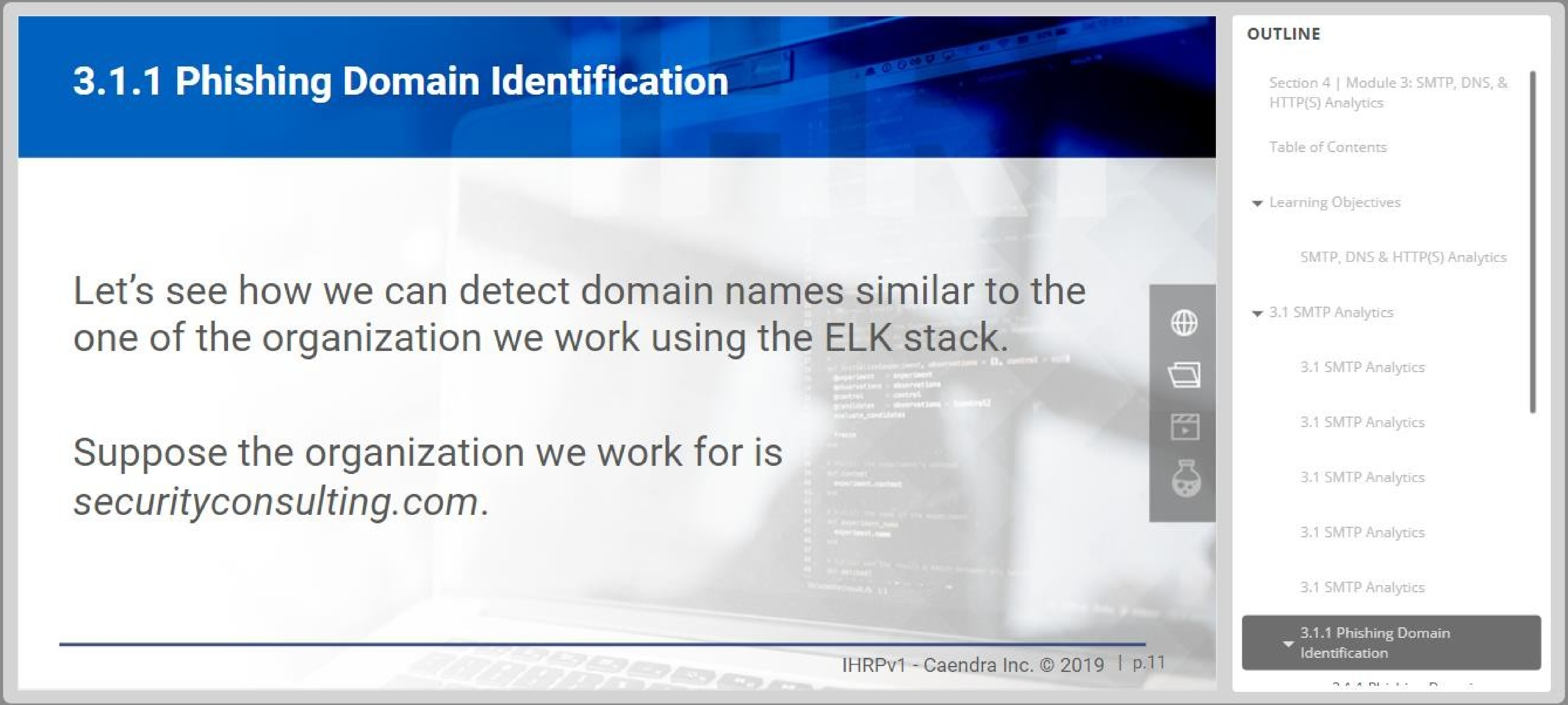
The majority of the aforementioned sources log fields such as *From*, *To*, *Subject*, *Reply Codes*, *Mail User Agent*, *Source IP*, *Destination IP*, *File attachment name*, *File attachment size* etc.



3.1 SMTP Analytics

To effectively perform SMTP analytics we should keep an eye for.

- Numerous e-mails being sent within a small time window from external sources
- Usage of key personnel names (possible whaling)
- Domain names similar to the one of the organization we work for
- E-mails being sent through unauthorized servers (if you notice a high volume of e-mails within a small time window to an online email provider this may indicate an e-mail-based Command and Control channel)
- Abnormal SMTP User Agents

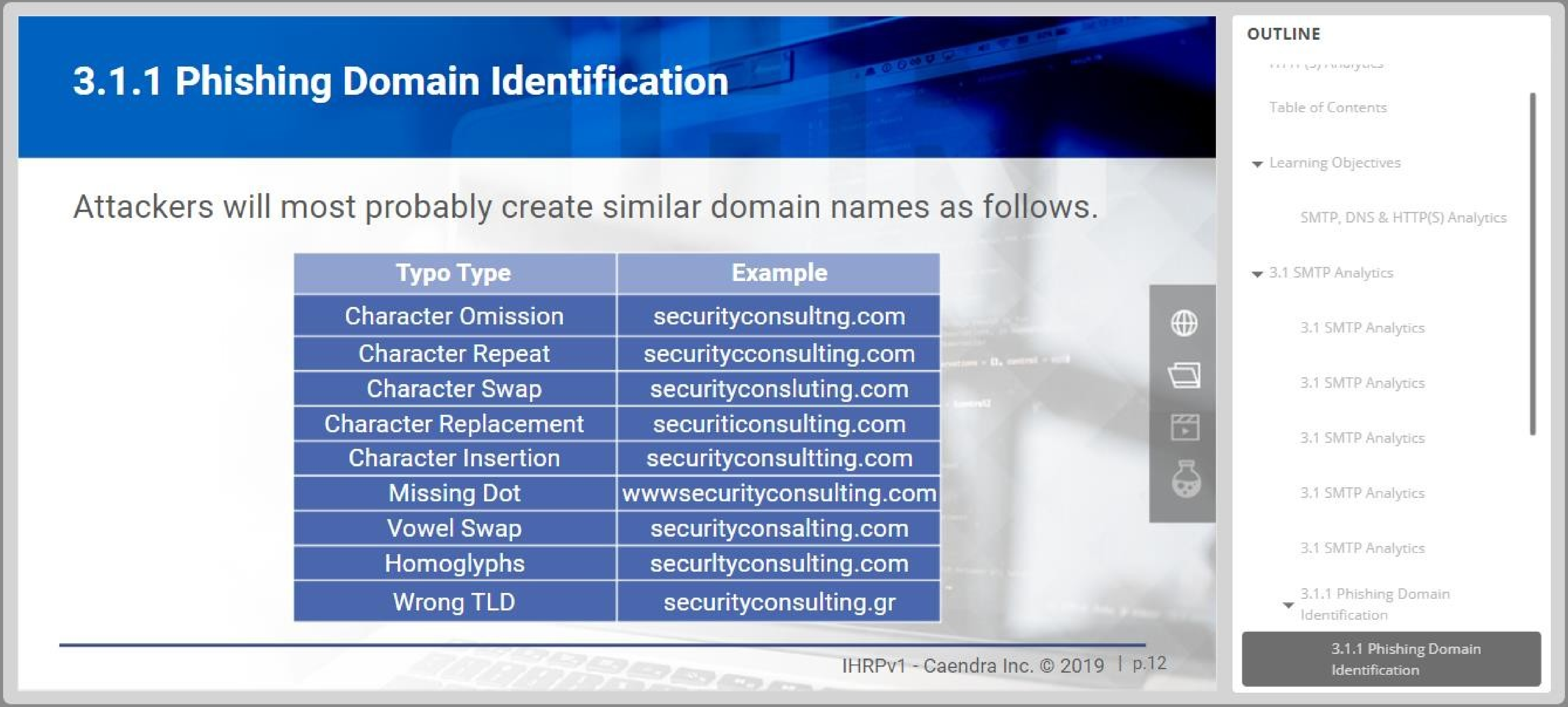


3.1.1 Phishing Domain Identification

Let’s see how we can detect domain names similar to the one of the organization we work using the ELK stack.

Suppose the organization we work for is *securityconsulting.com*.

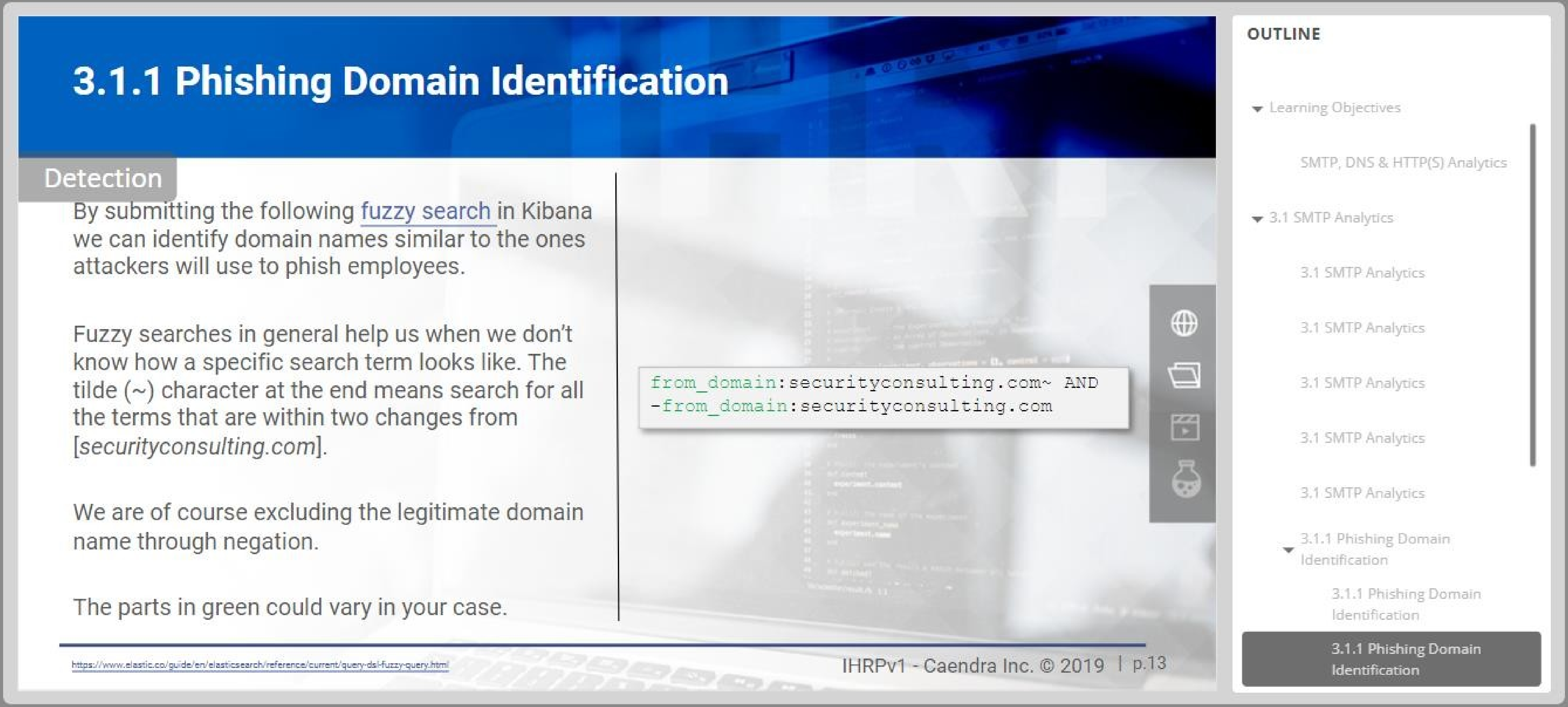
OUTLINE



3.1.1 Phishing Domain Identification

Attackers will most probably create similar domain names as follows.

Typo Type	Example
Character Omission	securityconsultng.com
Character Repeat	securitycconsulting.com
Character Swap	securityconsluting.com
Character Replacement	securiticonsulting.com
Character Insertion	securityconsultting.com
Missing Dot	wwwsecurityconsulting.com
Vowel Swap	securityconsalting.com
Homoglyphs	securltyconsulting.com
Wrong TLD	securityconsulting.gr


OUTLINE
Learning Objectives
SMTP, DNS & HTTP(S) Analytics
3.1 SMTP Analytics
3.1 SMTP Analytics
3.1 SMTP Analytics
3.1 SMTP Analytics
3.1 SMTP Analytics
3.1 SMTP Analytics
3.1.1 Phishing Domain Identification
3.1.1 Phishing Domain Identification
3.1.1 Phishing Domain Identification

3.1.1 Phishing Domain Identification

Detection

By submitting the following fuzzy search in Kibana we can identify domain names similar to the ones attackers will use to phish employees.

Fuzzy searches in general help us when we don't know how a specific search term looks like. The tilde (~) character at the end means search for all the terms that are within two changes from [*securityconsulting.com*].

We are of course excluding the legitimate domain name through negation.

The parts in green could vary in your case.

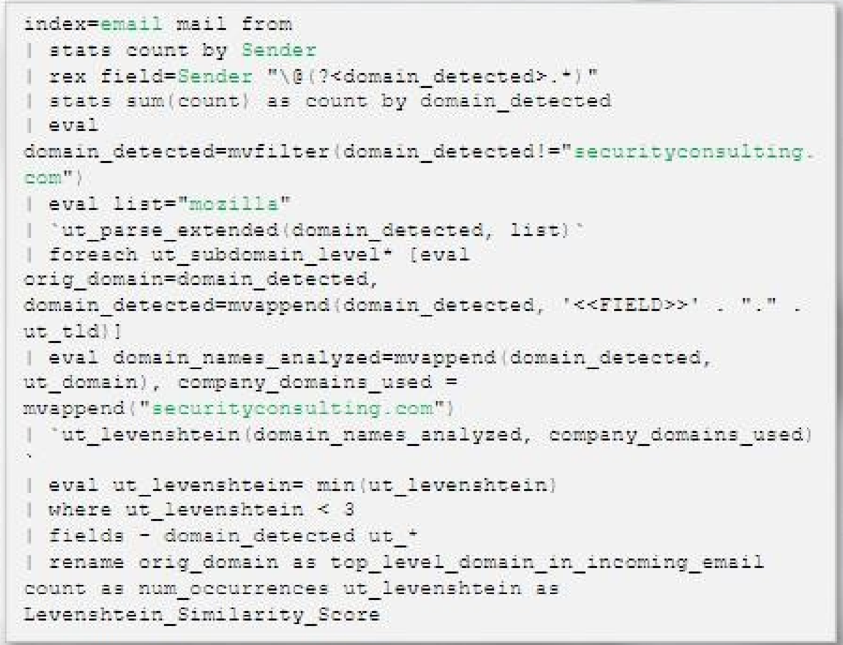
```
from_domain:securityconsulting.com~ AND
-from_domain:securityconsulting.com
```

https://www.elastic.co/guide/en/elasticsearch/reference/current/query-dsl-fuzzy-query.html

3.1.1 Phishing Domain Identification

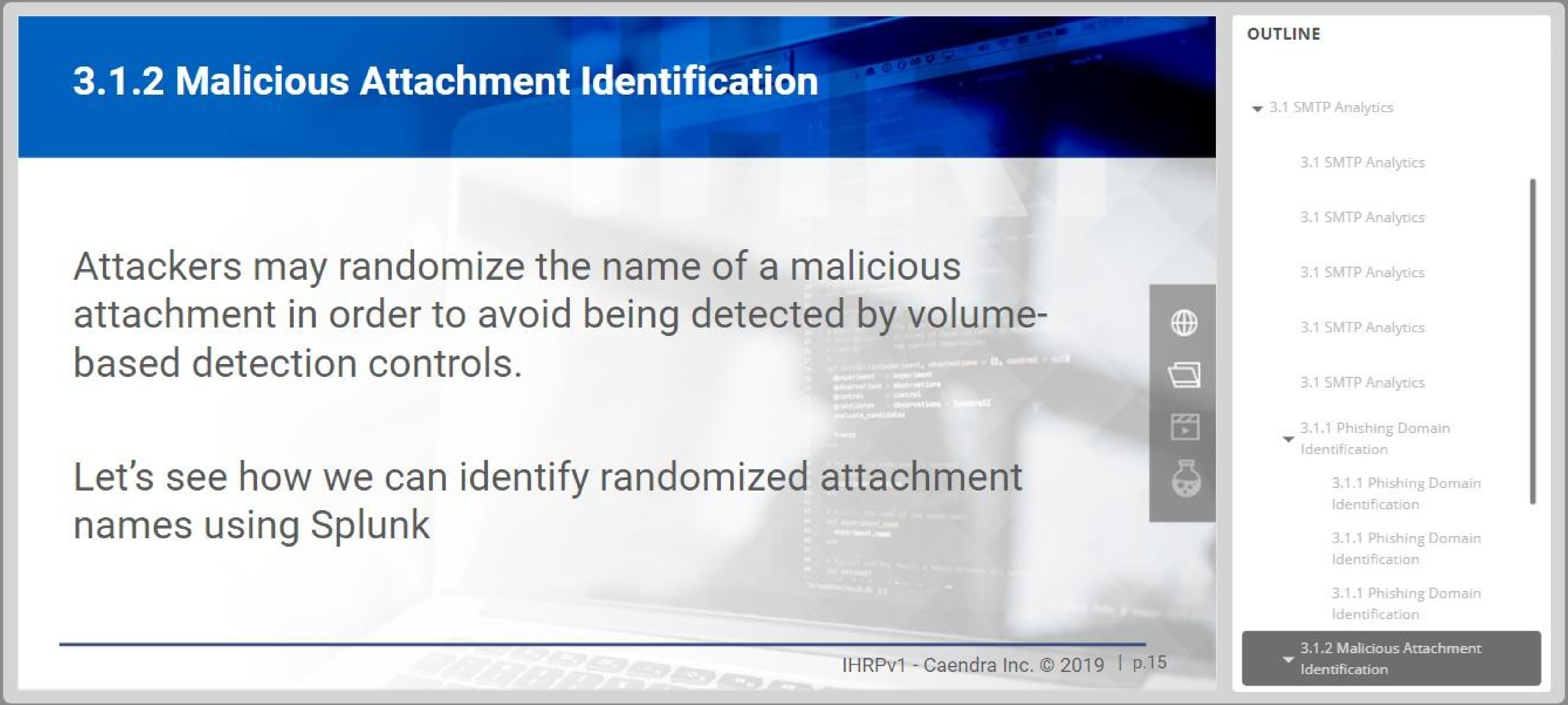
Detection

The same could have been achieved using Splunk and the Levenshtein distance.



```
index=email mail from
| stats count by Sender
| rex field=Sender "\@(?<domain_detected>.*)"
| stats sum(count) as count by domain_detected
| eval
domain_detected=mvfilter(domain_detected!="securityconsulting.
com")
| eval list="mozilla"
| `ut_parse_extended(domain_detected, list)`
| foreach ut_subdomain_level* [eval
orig_domain=domain_detected,
domain_detected=mvappend(domain_detected, '<<FIELD>>' . "." .
ut_tld)]
| eval domain_names_analyzed=mvappend(domain_detected,
ut_domain), company_domains_used =
mvappend("securityconsulting.com")
| `ut_levenshtein(domain_names_analyzed, company_domains_used)
`
| eval ut_levenshtein= min(ut_levenshtein)
| where ut_levenshtein < 3
| fields - domain_detected ut_*
| rename orig_domain as top_level_domain_in_incoming_email
count as num_occurrences ut_levenshtein as
Levenshtein_Similarity_Score
```

https://www.splunk.com/blog/2017/11/03/you-can-t-hyde-from-dr-levenshtein-when-you-use-url-toolbox.html

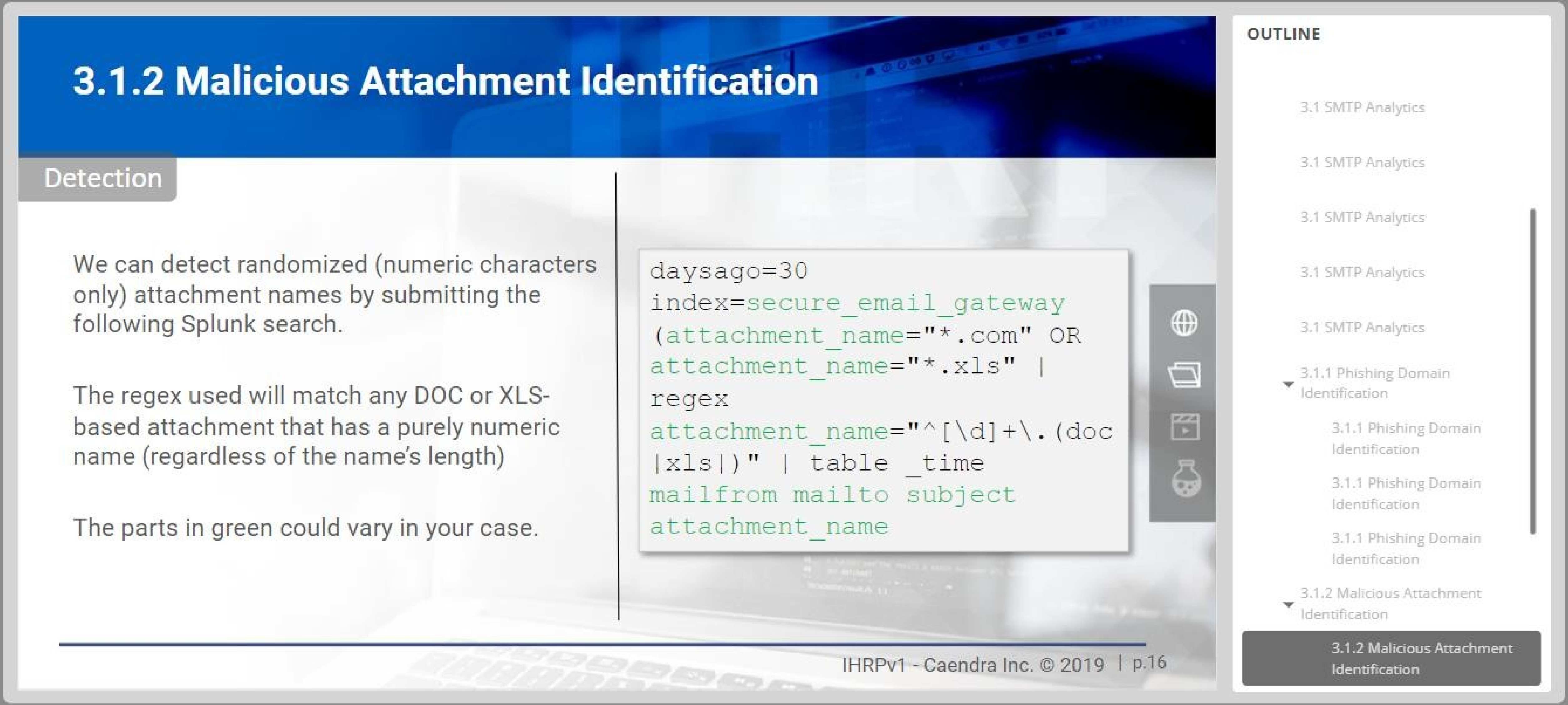


3.1.2 Malicious Attachment Identification

Attackers may randomize the name of a malicious attachment in order to avoid being detected by volume-based detection controls.

Let's see how we can identify randomized attachment names using Splunk

OUTLINE
3.1 SMTP Analytics
3.1 SMTP Analytics
3.1 SMTP Analytics
3.1 SMTP Analytics
3.1 SMTP Analytics
3.1 SMTP Analytics
3.1.1 Phishing Domain Identification
3.1.1 Phishing Domain Identification
3.1.1 Phishing Domain Identification
3.1.1 Phishing Domain Identification
3.1.2 Malicious Attachment Identification



3.1.2 Malicious Attachment Identification

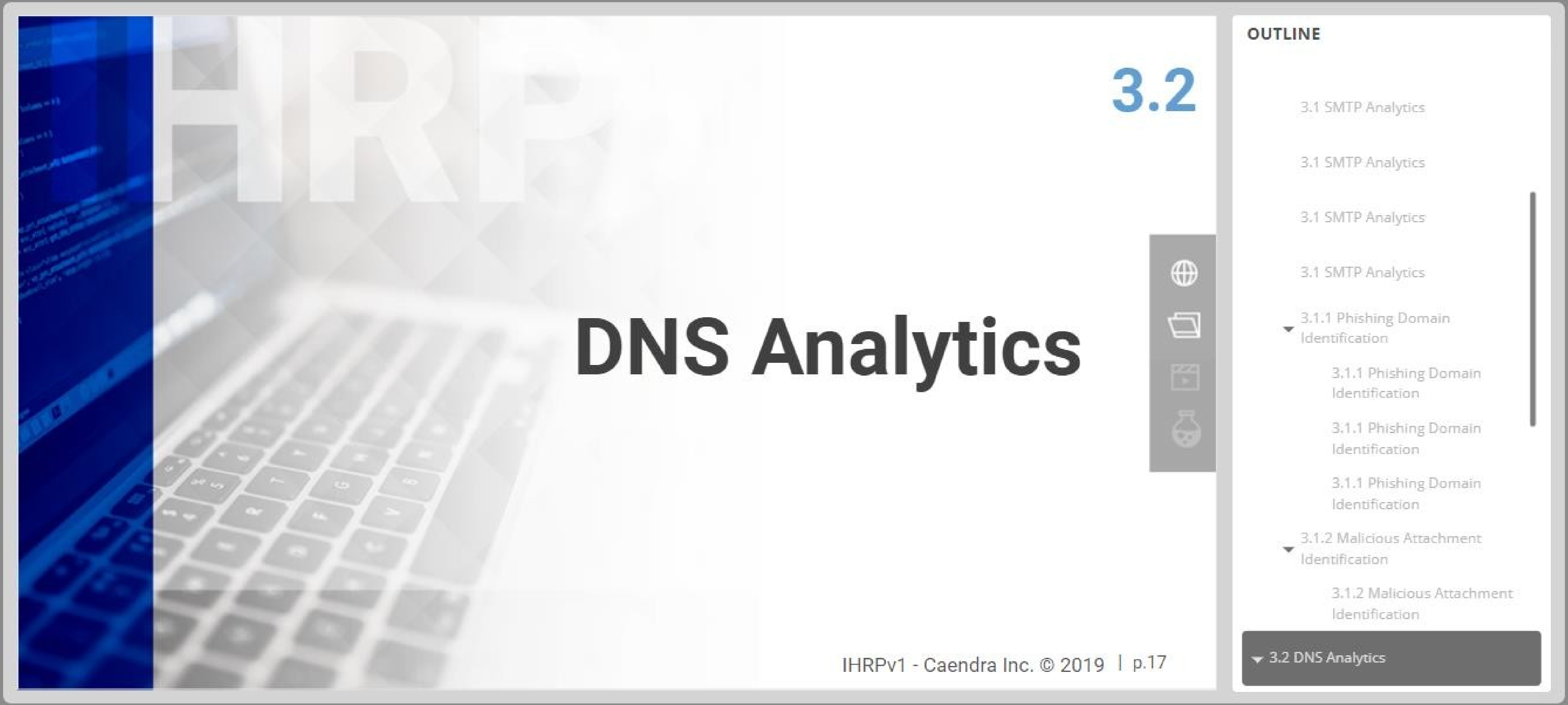
Detection

We can detect randomized (numeric characters only) attachment names by submitting the following Splunk search.

The regex used will match any DOC or XLS-based attachment that has a purely numeric name (regardless of the name's length)

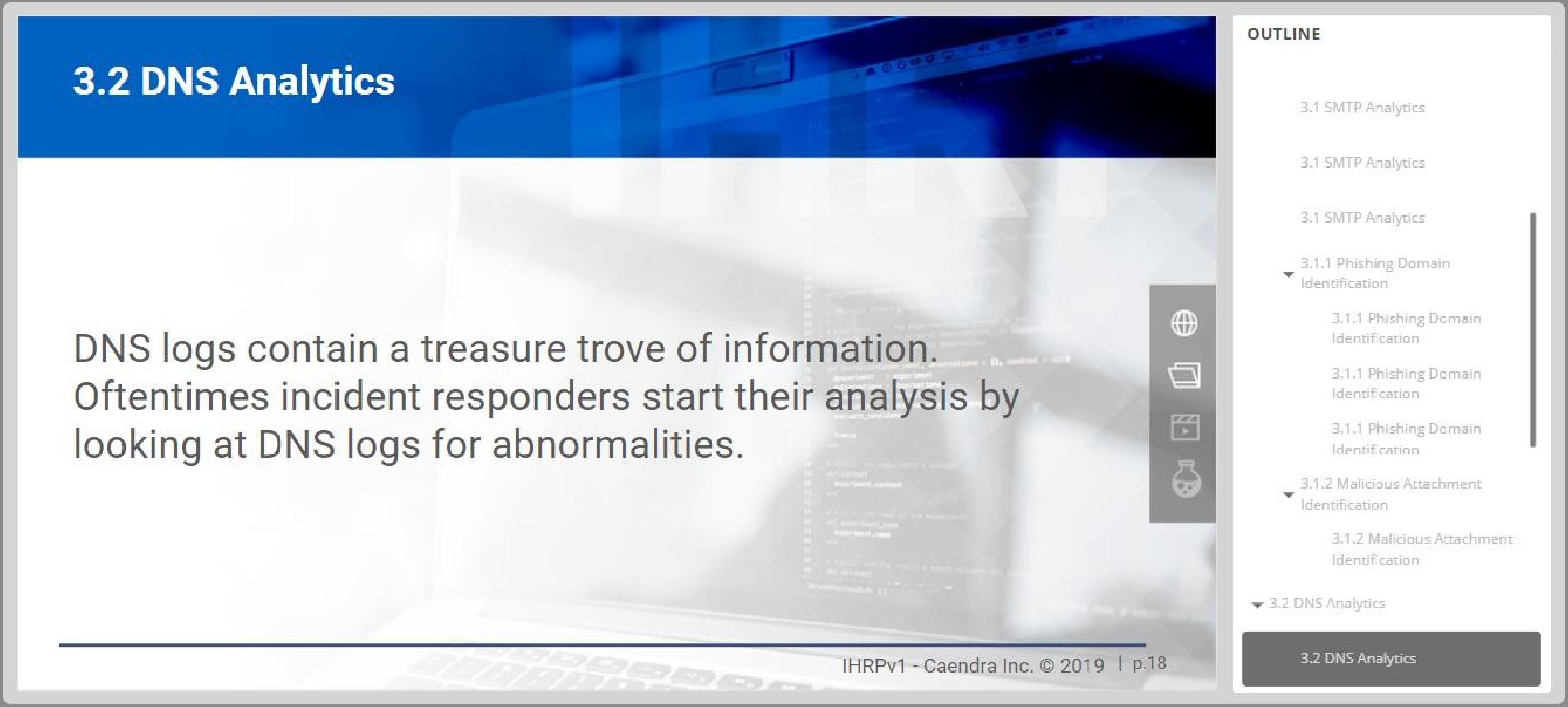
The parts in green could vary in your case.

```
daysago=30
index=secure_email_gateway
(attachment_name="*.com" OR
attachment_name="*.xls" |
regex
attachment_name="^[\d]+\.(doc
|xls|)" | table _time
mailfrom mailto subject
attachment_name
```


OUTLINE
3.1 SMTP Analytics
3.1 SMTP Analytics
3.1 SMTP Analytics
3.1 SMTP Analytics
3.1.1 Phishing Domain Identification
3.1.1 Phishing Domain Identification
3.1.1 Phishing Domain Identification
3.1.1 Phishing Domain Identification
3.1.2 Malicious Attachment Identification
3.1.2 Malicious Attachment Identification
3.2 DNS Analytics

3.2

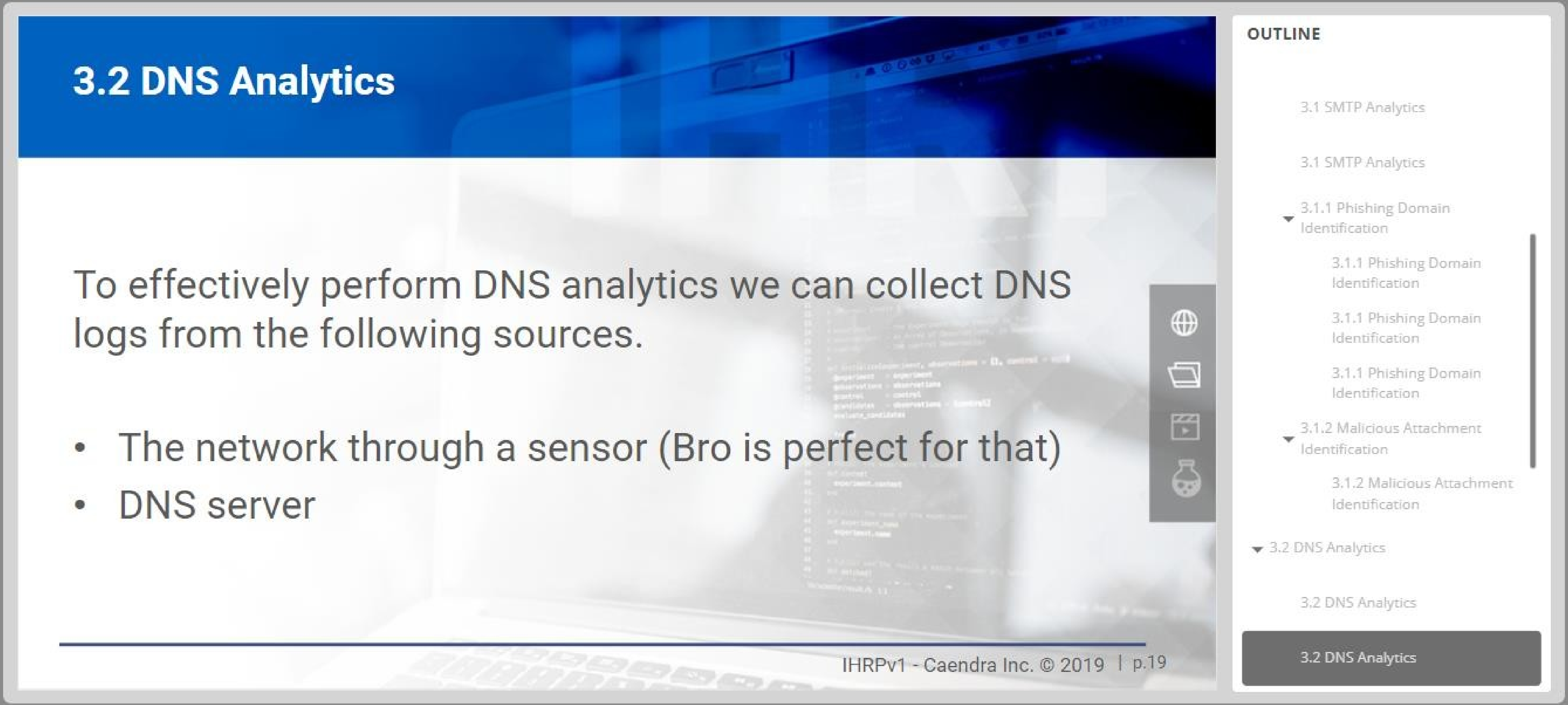
DNS Analytics



3.2 DNS Analytics

DNS logs contain a treasure trove of information. Oftentimes incident responders start their analysis by looking at DNS logs for abnormalities.

OUTLINE

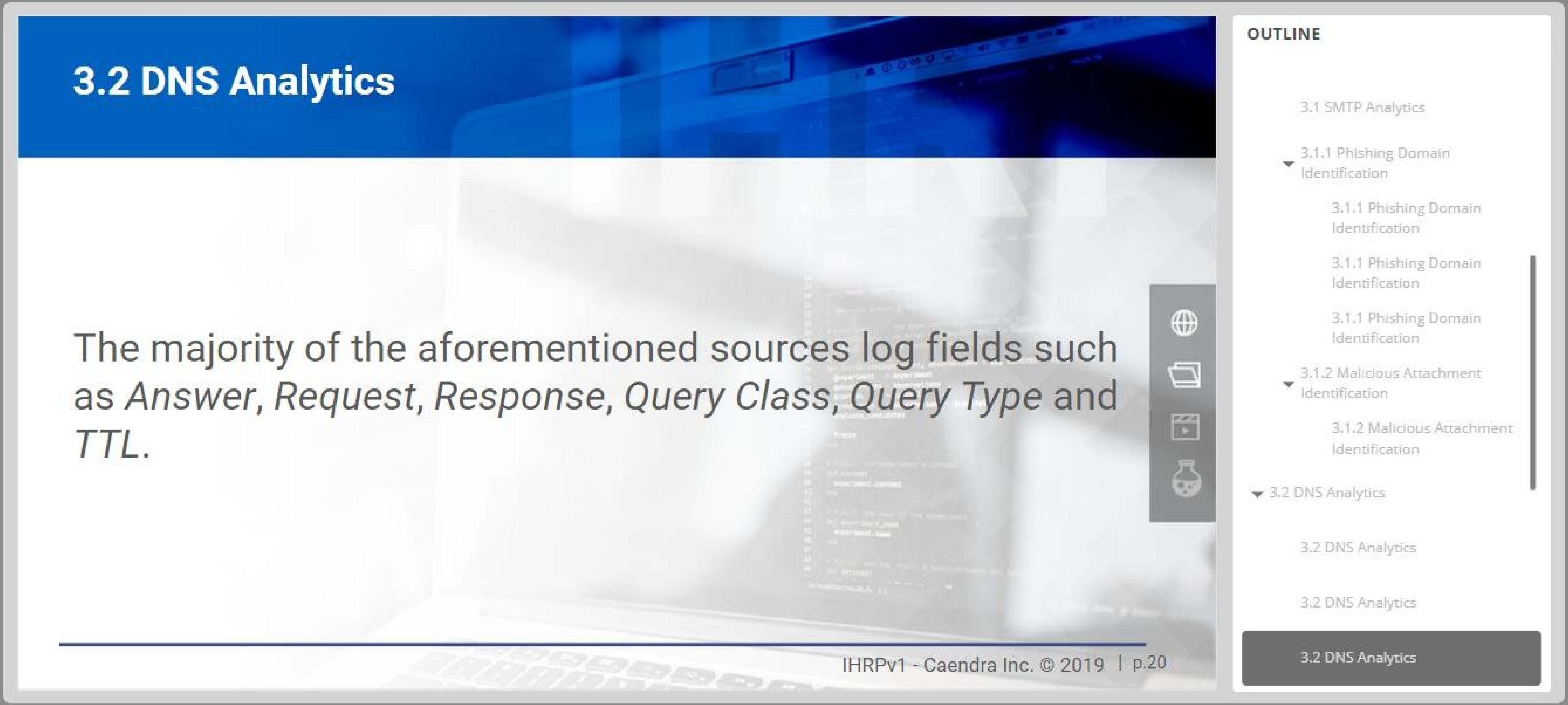


3.2 DNS Analytics

To effectively perform DNS analytics we can collect DNS logs from the following sources.

- The network through a sensor (Bro is perfect for that)
- DNS server

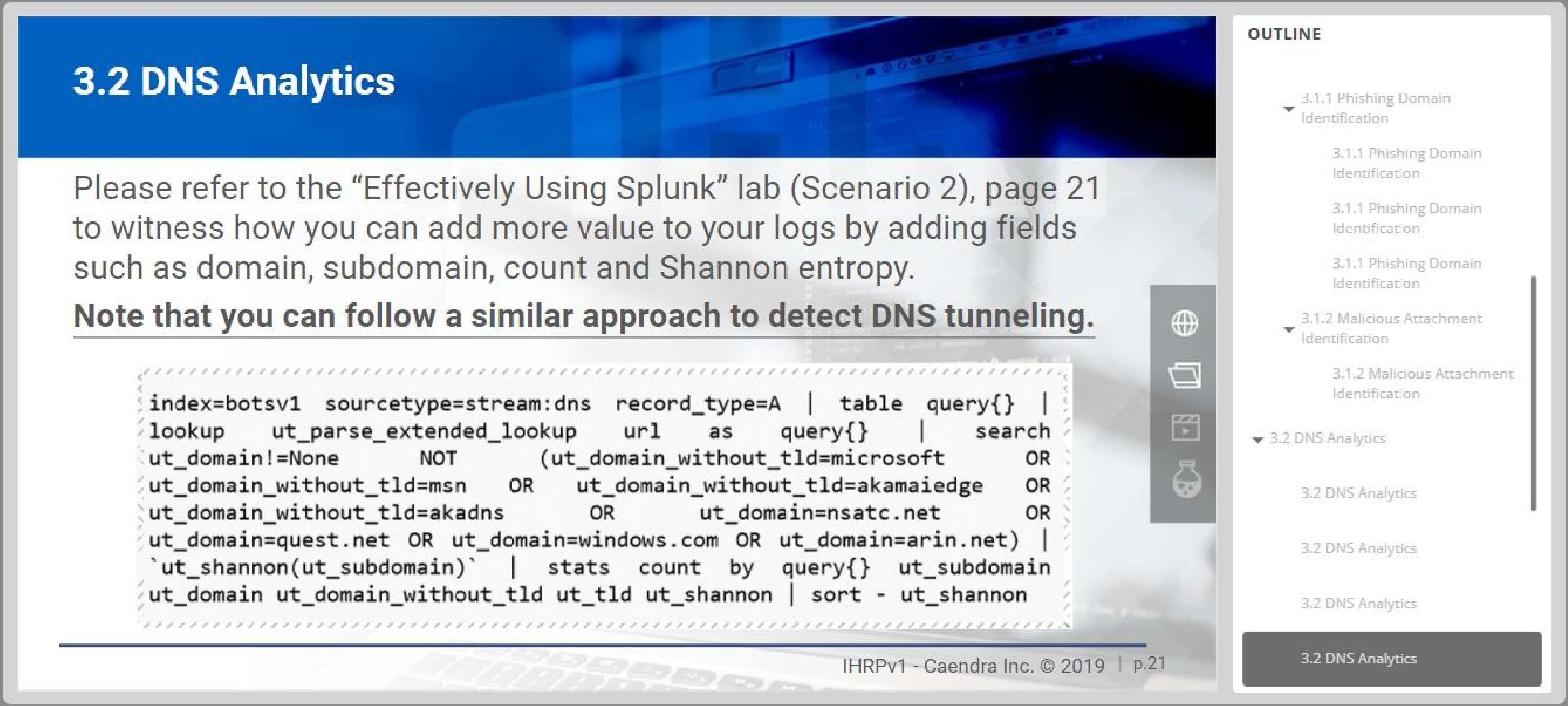
OUTLINE



3.2 DNS Analytics

The majority of the aforementioned sources log fields such as *Answer*, *Request*, *Response*, *Query Class*, *Query Type* and *TTL*.

OUTLINE



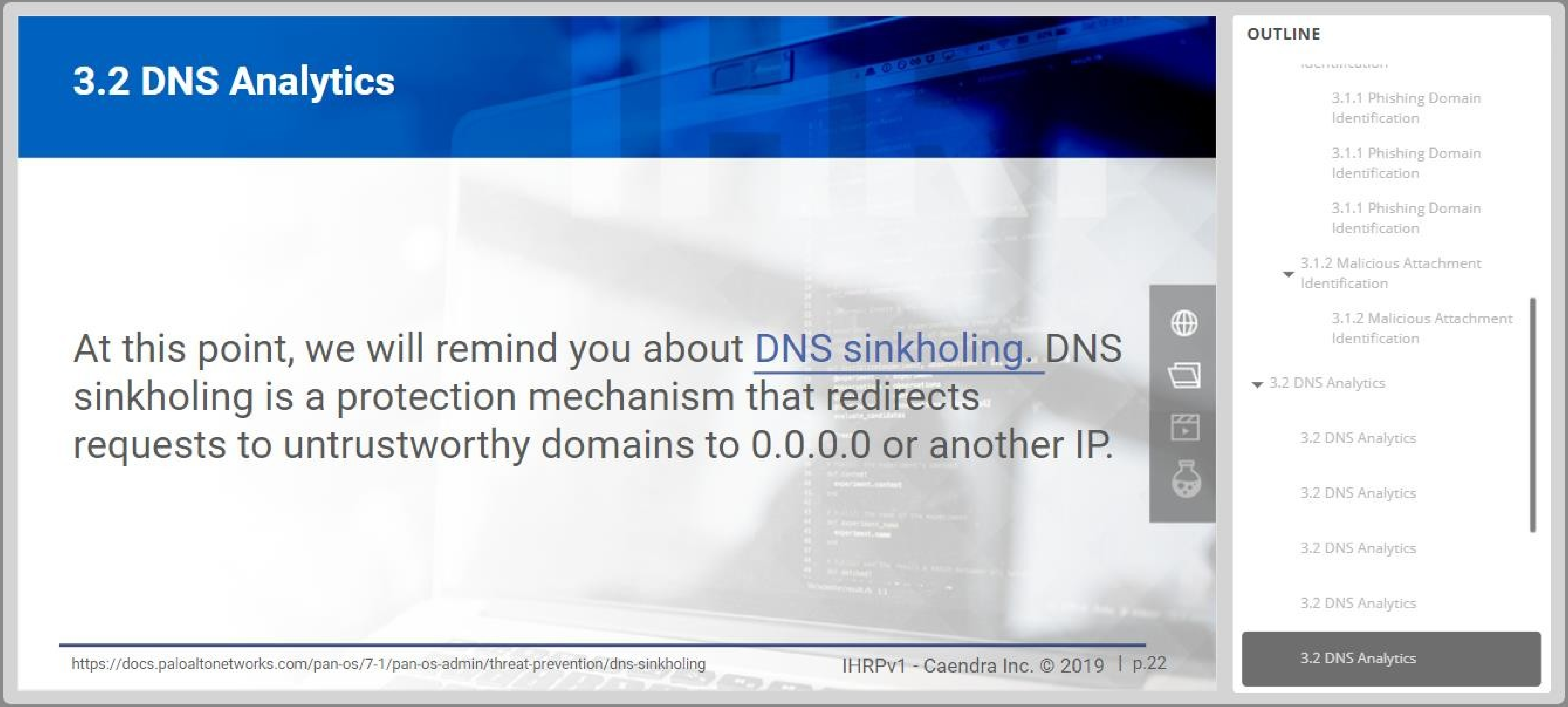
3.2 DNS Analytics

Please refer to the “Effectively Using Splunk” lab (Scenario 2), page 21 to witness how you can add more value to your logs by adding fields such as domain, subdomain, count and Shannon entropy.

Note that you can follow a similar approach to detect DNS tunneling.

```
index=botsv1 sourcetype=stream:dns record_type=A | table query{} |
lookup ut_parse_extended_lookup url as query{} | search
ut_domain!=None NOT (ut_domain_without_tld=microsoft OR
ut_domain_without_tld=msn OR ut_domain_without_tld=akamaiedge OR
ut_domain_without_tld=akadns OR ut_domain=nsatc.net OR
ut_domain=quest.net OR ut_domain=windows.com OR ut_domain=arin.net) |
`ut_shannon(ut_subdomain)` | stats count by query{} ut_subdomain
ut_domain ut_domain_without_tld ut_tld ut_shannon | sort - ut_shannon
```

OUTLINE

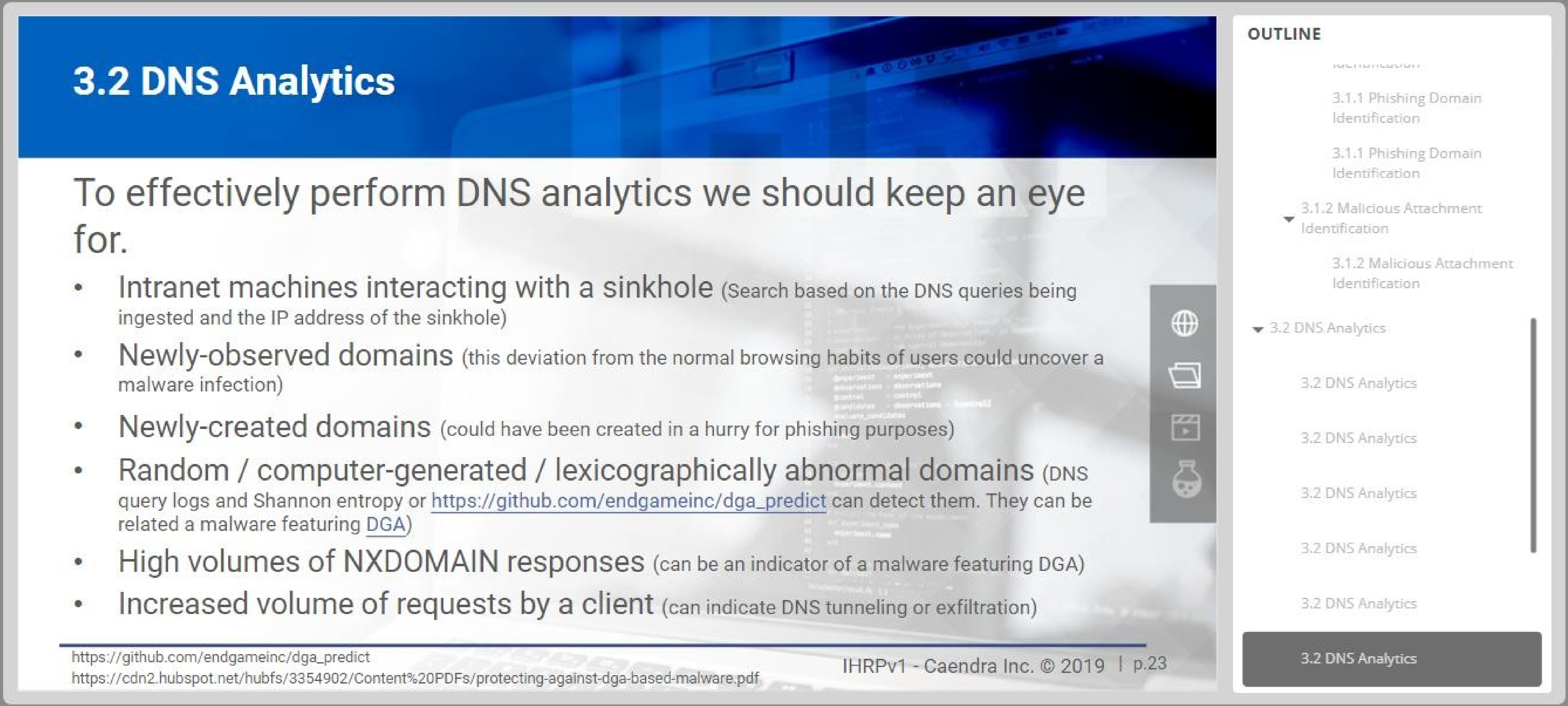


OUTLINE
3.1.1 Phishing Domain Identification
3.1.1 Phishing Domain Identification
3.1.1 Phishing Domain Identification
3.1.2 Malicious Attachment Identification
3.1.2 Malicious Attachment Identification
3.2 DNS Analytics
3.2 DNS Analytics
3.2 DNS Analytics
3.2 DNS Analytics
3.2 DNS Analytics
3.2 DNS Analytics

3.2 DNS Analytics

At this point, we will remind you about DNS sinkholing. DNS sinkholing is a protection mechanism that redirects requests to untrustworthy domains to 0.0.0.0 or another IP.

https://docs.paloaltonetworks.com/pan-os/7-1/pan-os-admin/threat-prevention/dns-sinkholing



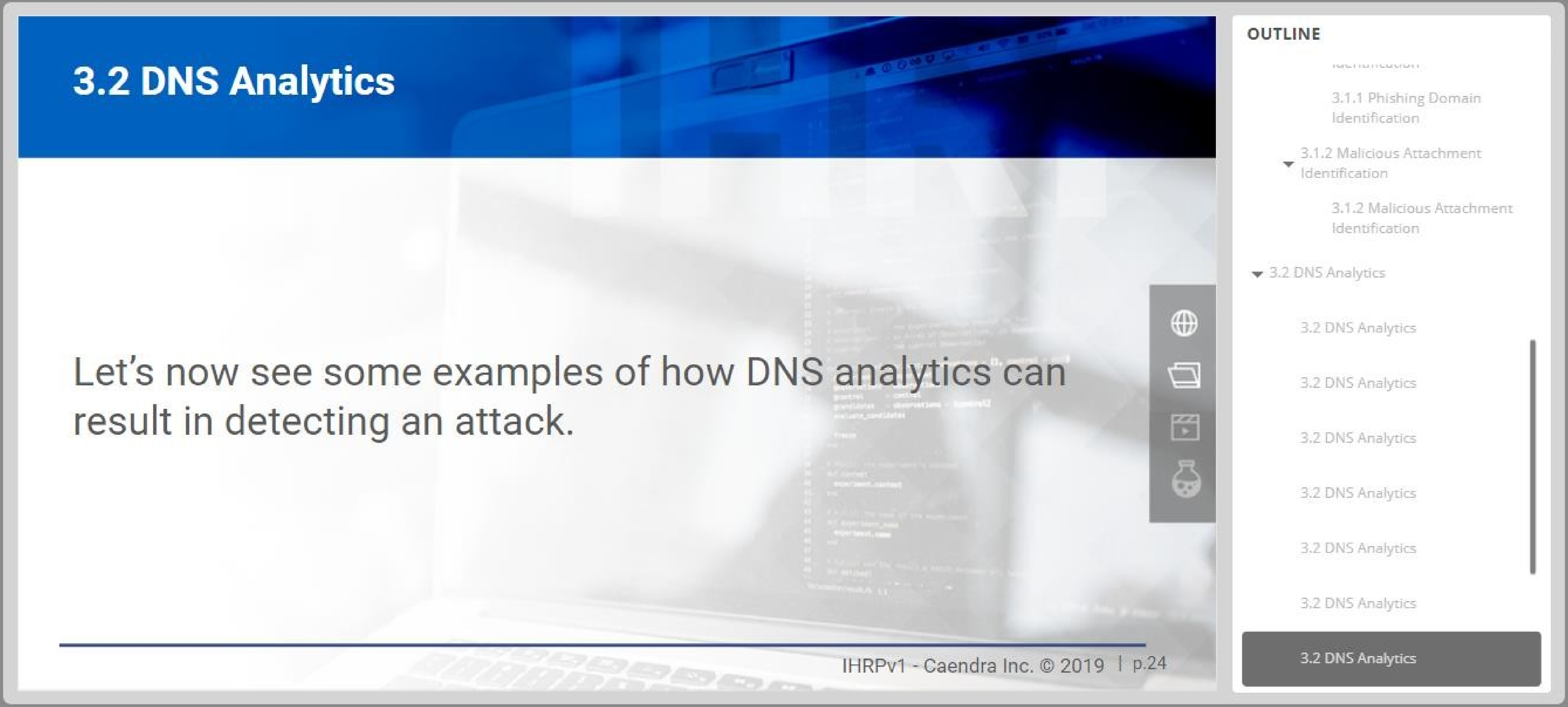
OUTLINE
Identification
3.1.1 Phishing Domain Identification
3.1.1 Phishing Domain Identification
3.1.2 Malicious Attachment Identification
3.1.2 Malicious Attachment Identification
3.2 DNS Analytics
3.2 DNS Analytics
3.2 DNS Analytics
3.2 DNS Analytics
3.2 DNS Analytics
3.2 DNS Analytics
3.2 DNS Analytics

3.2 DNS Analytics

To effectively perform DNS analytics we should keep an eye for.

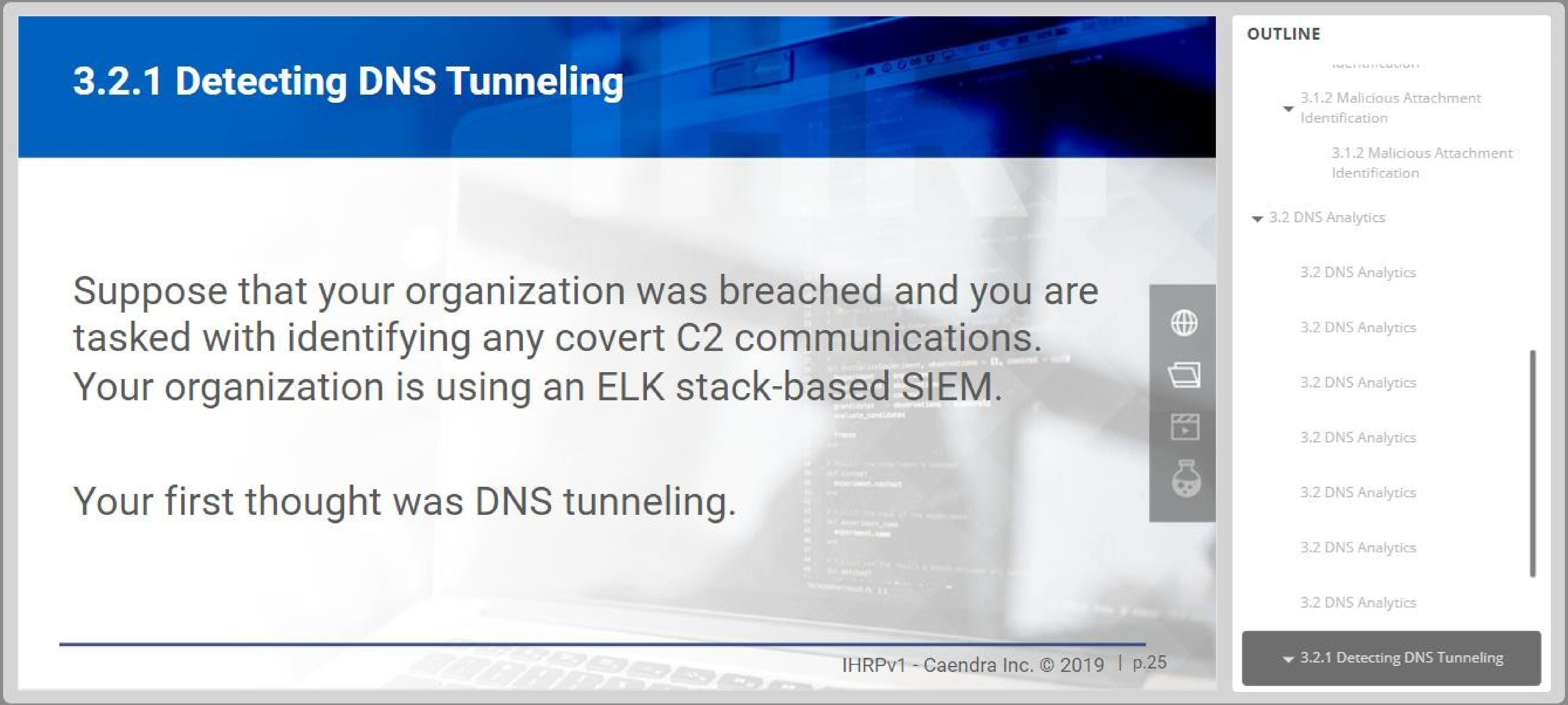
- Intranet machines interacting with a sinkhole (Search based on the DNS queries being ingested and the IP address of the sinkhole)
- Newly-observed domains (this deviation from the normal browsing habits of users could uncover a malware infection)
- Newly-created domains (could have been created in a hurry for phishing purposes)
- Random / computer-generated / lexicographically abnormal domains (DNS query logs and Shannon entropy or https://github.com/endgameinc/dga_predict can detect them. They can be related a malware featuring DGA)
- High volumes of NXDOMAIN responses (can be an indicator of a malware featuring DGA)
- Increased volume of requests by a client (can indicate DNS tunneling or exfiltration)

https://github.com/endgameinc/dga_predict
https://cdn2.hubspot.net/hubfs/3354902/Content%20PDFs/protecting-against-dga-based-malware.pdf



3.2 DNS Analytics

Let’s now see some examples of how DNS analytics can result in detecting an attack.



3.2.1 Detecting DNS Tunneling

Suppose that your organization was breached and you are tasked with identifying any covert C2 communications. Your organization is using an ELK stack-based SIEM.

Your first thought was DNS tunneling.

3.2.1 Detecting DNS Tunneling

Detection

Instead of a search, this time let's see a visualization that can detect DNS tunneling.

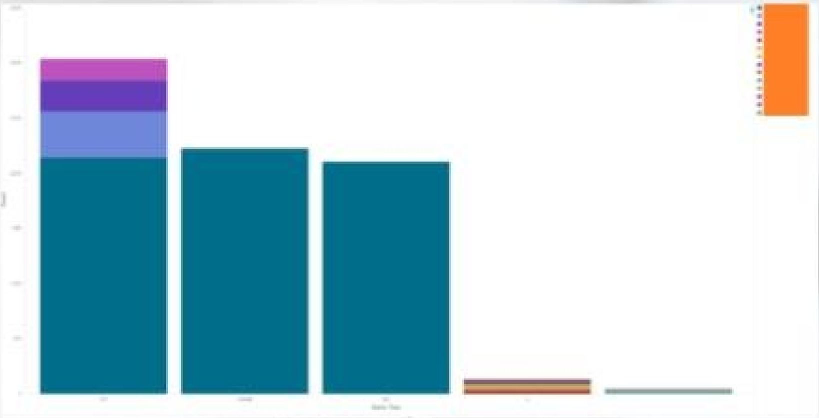
The idea is to visualize DNS query type by domain.

1. Visualize -> Create a new visualization -> Vertical chart bar
2. Click "From a new search" and select the index pattern that contains the DNS logs
3. Click on *X-Axis*, on the *Aggregation* drop down menu choose *Terms*, on the *Field* drop down menu choose *query_type_name.raw* and set the *CustomLabel* to *Query Type*. Then, click on *Add sub-buckets* and select *Split Bars* for bucket type. Select *Terms* for *Sub Aggregation* and highest_registered_domain* for Field. Finally, set the *CustomLabel* to *Domain*.

The parts in green could vary in your case.

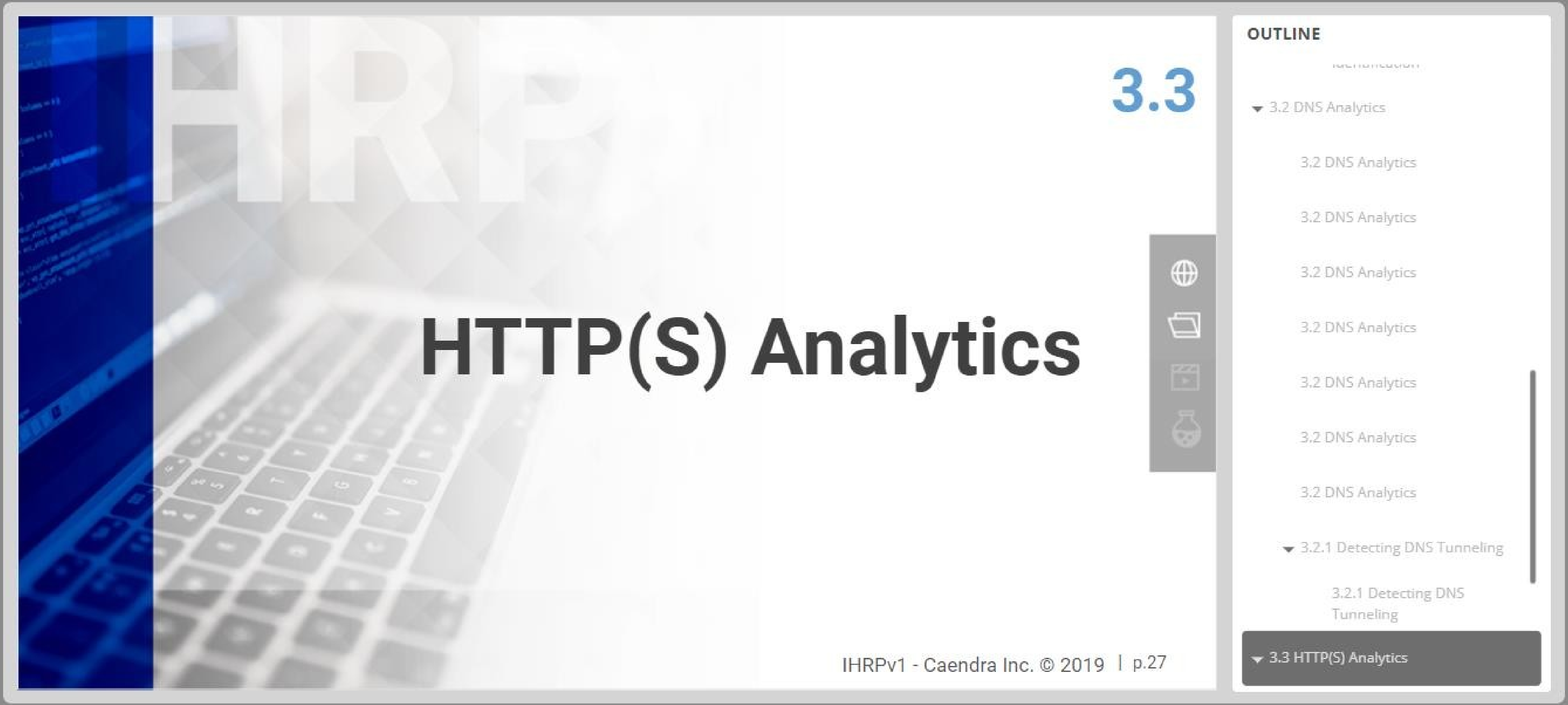
It is quite obvious that the domain depicted in petrol color has a vast number of TXT, CNAME and MX records compared to the other domains. We are most probably dealing with a DNS tunneling attack.

Splunk can also easily detect DNS tunneling as follows.
https://www.reddit.com/r/netsec/comments/4aco2v/detect_dns_tunneling_done_by_tools_such_as_iodine/



* This analysis requires fields that contain the highest registered domain (domain) of each DNS query. If these fields are not available in the source data, a transformation must be applied to create them.

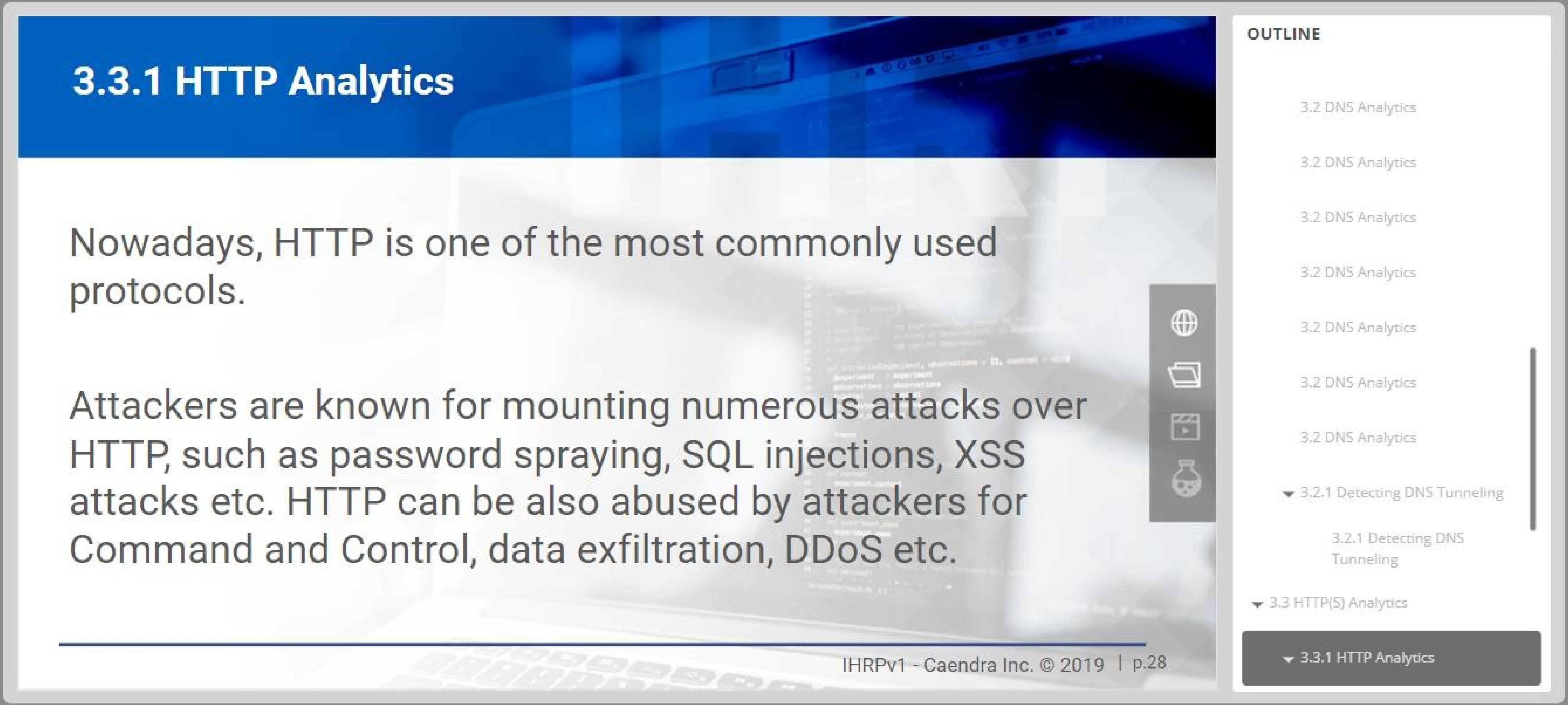
https://www.elastic.co/products/stack/machine-learning/recipes/dns-data-exfiltration-tunneling
https://www.reddit.com/r/netsec/comments/4aco2v/detect_dns_tunneling_done_by_tools_such_as_iodine/



3.3

HTTP(S) Analytics

OUTLINE

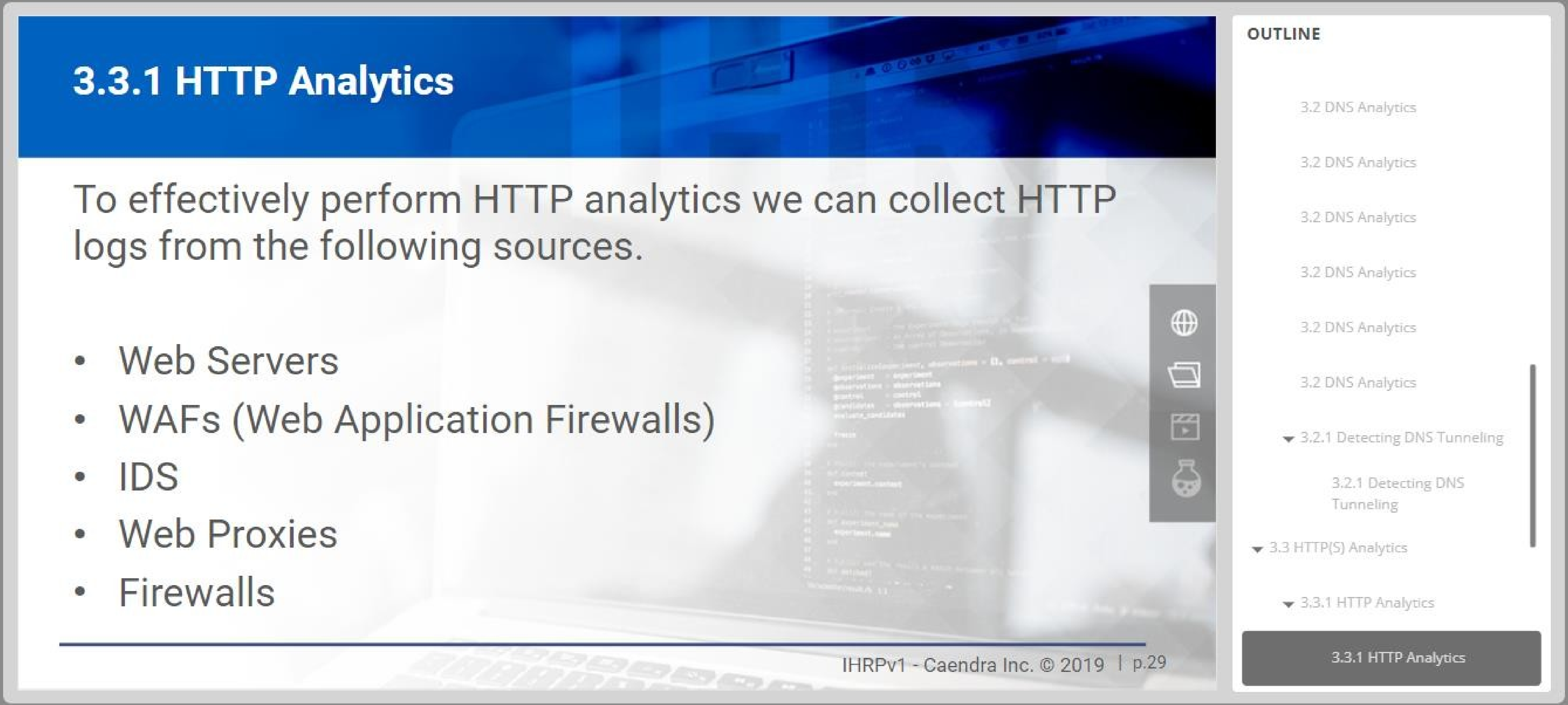


3.3.1 HTTP Analytics

Nowadays, HTTP is one of the most commonly used protocols.

Attackers are known for mounting numerous attacks over HTTP, such as password spraying, SQL injections, XSS attacks etc. HTTP can be also abused by attackers for Command and Control, data exfiltration, DDoS etc.

OUTLINE



3.3.1 HTTP Analytics

To effectively perform HTTP analytics we can collect HTTP logs from the following sources.

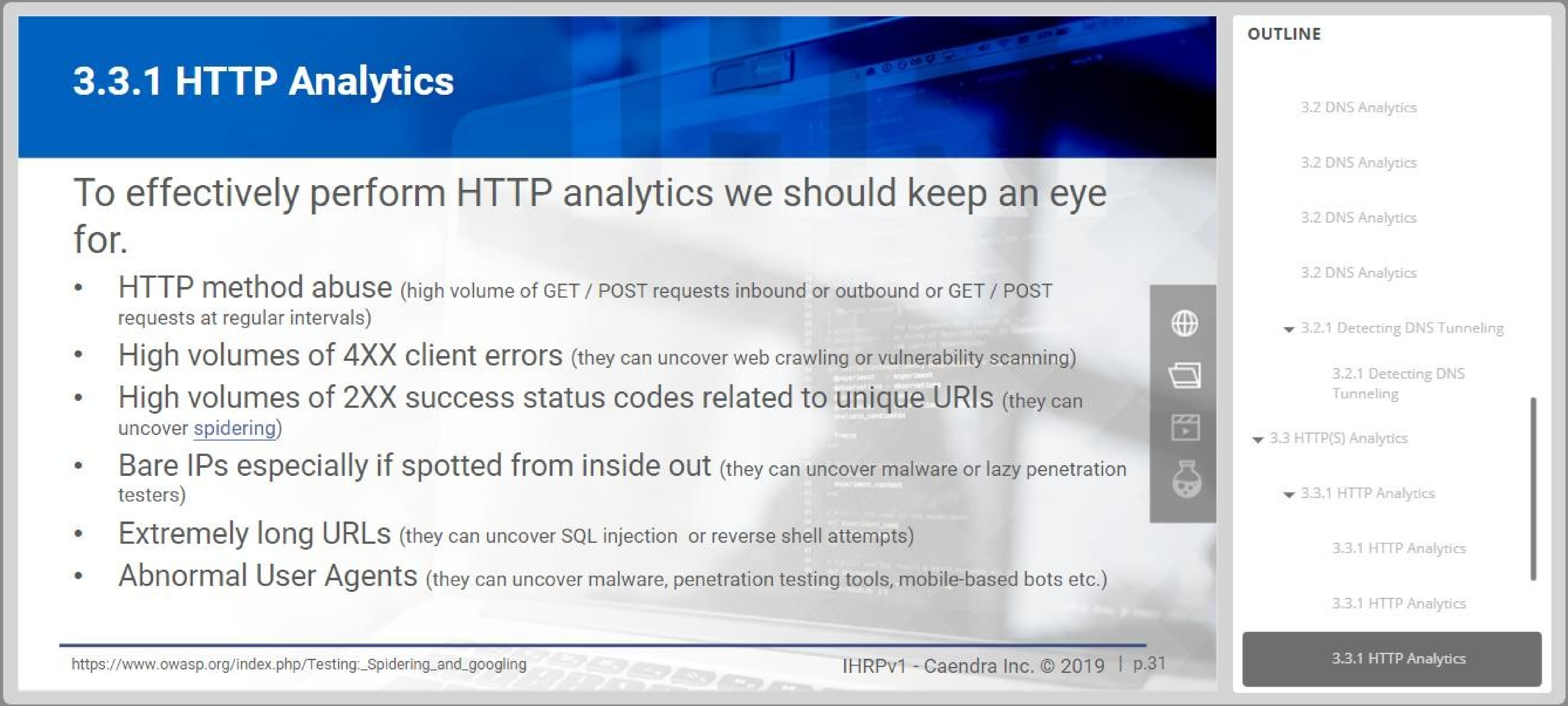
- Web Servers
- WAFs (Web Application Firewalls)
- IDS
- Web Proxies
- Firewalls

OUTLINE

3.3.1 HTTP Analytics

The majority of the aforementioned sources log fields such as *Timestamp*, *Source IP*, *Source Port*, *Destination IP*, *Destination Port*, *Method*, *Virtual Host*, *Referer*, *URI*, *User-Agent*, *Request Bytes*, *Response Bytes*, *Status Code*, *User*, *Proxy*, *Server Name*, *Duration*, *Cookie* and *MIME type*.

OUTLINE



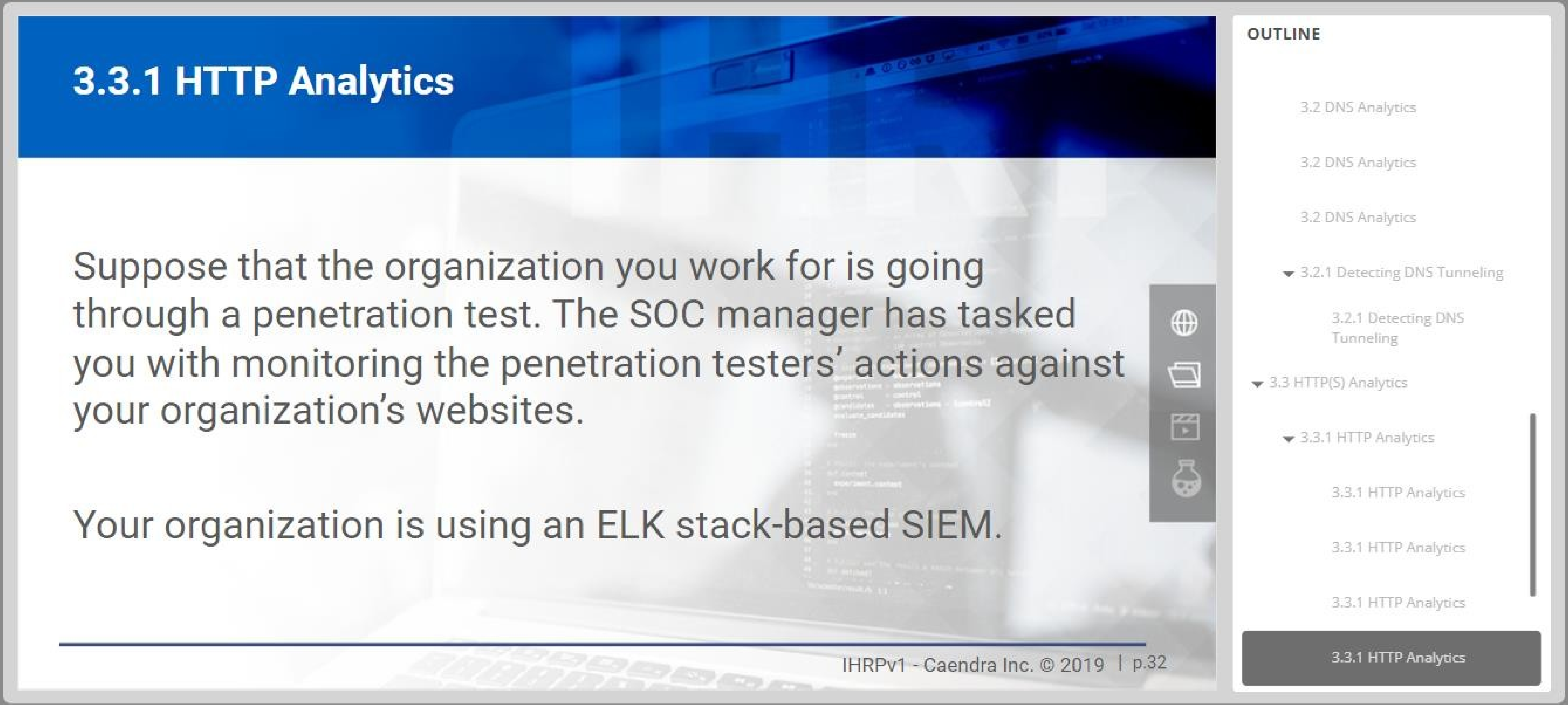
3.3.1 HTTP Analytics

To effectively perform HTTP analytics we should keep an eye for.

- HTTP method abuse (high volume of GET / POST requests inbound or outbound or GET / POST requests at regular intervals)
- High volumes of 4XX client errors (they can uncover web crawling or vulnerability scanning)
- High volumes of 2XX success status codes related to unique URIs (they can uncover spidering)
- Bare IPs especially if spotted from inside out (they can uncover malware or lazy penetration testers)
- Extremely long URLs (they can uncover SQL injection or reverse shell attempts)
- Abnormal User Agents (they can uncover malware, penetration testing tools, mobile-based bots etc.)

https://www.owasp.org/index.php/Testing:_Spidering_and_googling

OUTLINE



3.3.1 HTTP Analytics

Suppose that the organization you work for is going through a penetration test. The SOC manager has tasked you with monitoring the penetration testers' actions against your organization's websites.

Your organization is using an ELK stack-based SIEM.

3.3.1 HTTP Analytics

Detection

Let’s start by focusing on 404 status codes, that may indicate the usage of a vulnerability scanner or web crawler. We can do that by submitting the search you see on your right (upper image).

We can also focus on 200 status codes, that may indicate spidering activities. We can do that by submitting the search you see on your right (image at the bottom).

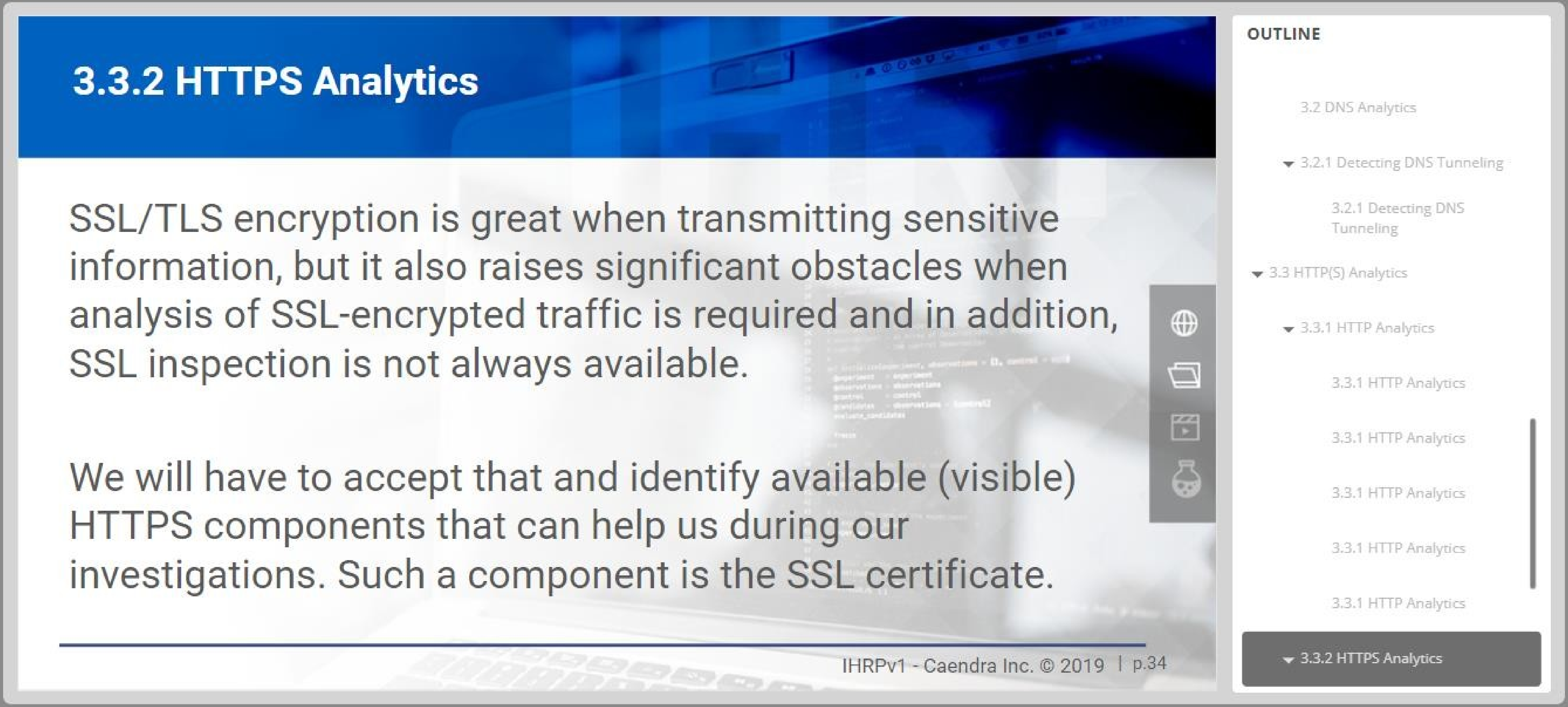
By analyzing the returned results of the first search, we identified the following.

The penetration testers were using the Nessus vulnerability scanner.

```
type:http AND status_code:404
```

```
type:http AND status_code:200
```

OUTLINE

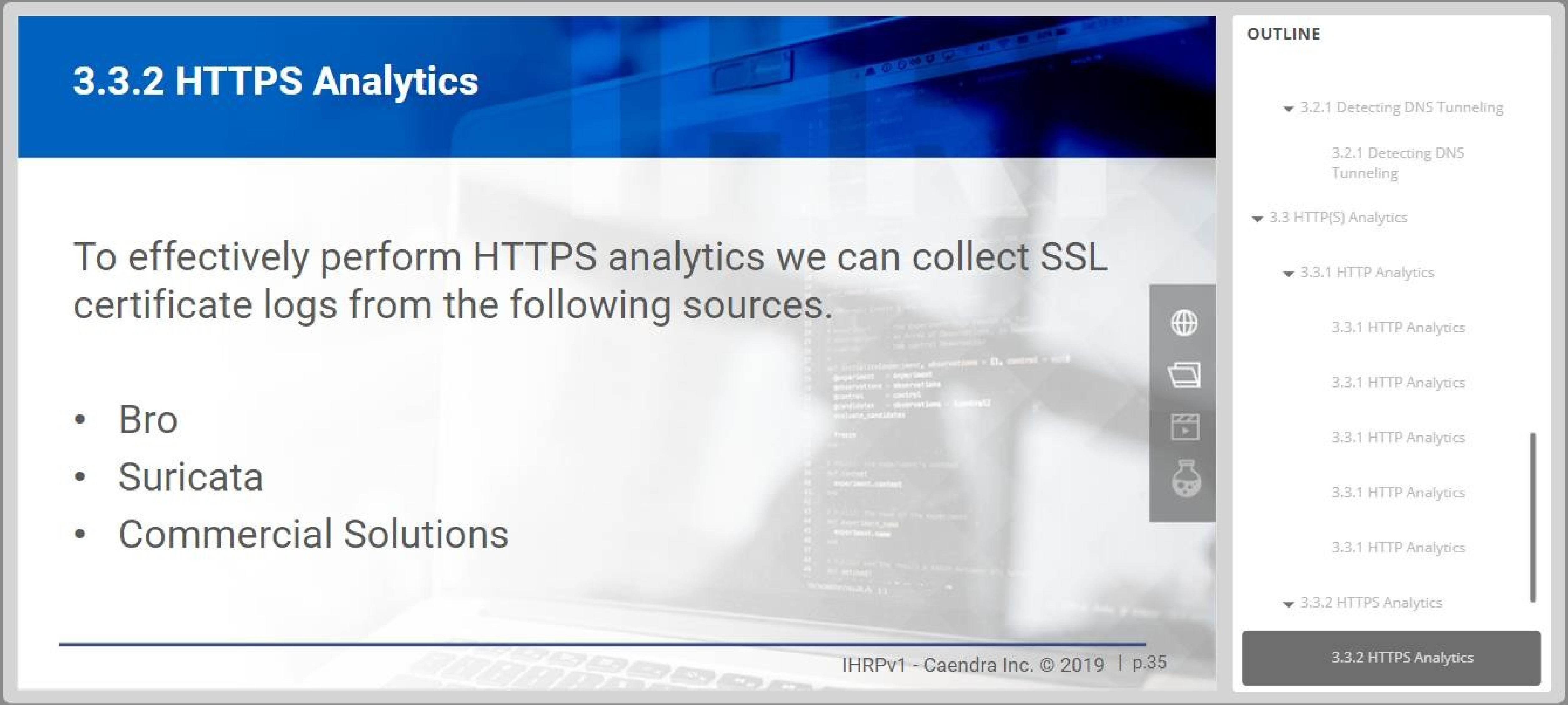


3.3.2 HTTPS Analytics

SSL/TLS encryption is great when transmitting sensitive information, but it also raises significant obstacles when analysis of SSL-encrypted traffic is required and in addition, SSL inspection is not always available.

We will have to accept that and identify available (visible) HTTPS components that can help us during our investigations. Such a component is the SSL certificate.

OUTLINE

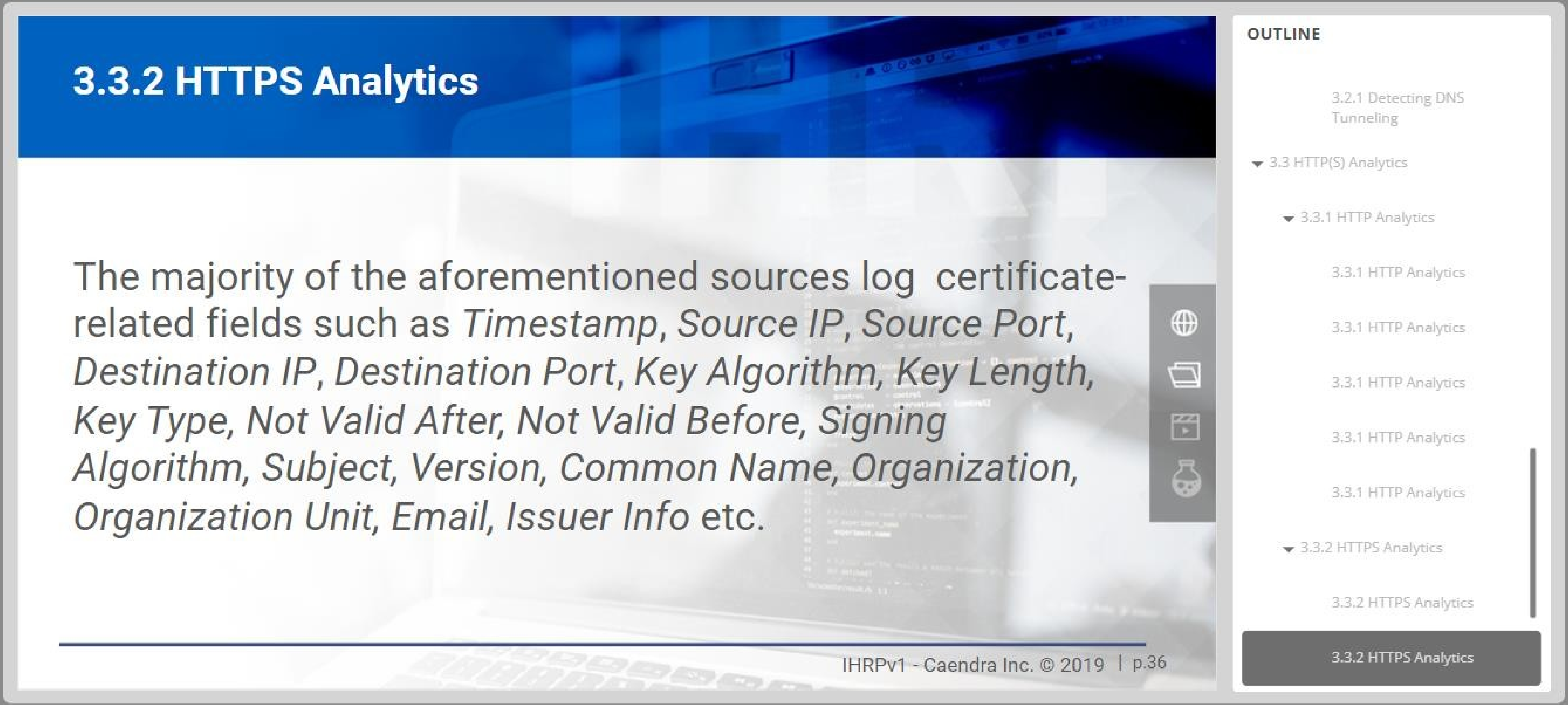


3.3.2 HTTPS Analytics

To effectively perform HTTPS analytics we can collect SSL certificate logs from the following sources.

- Bro
- Suricata
- Commercial Solutions

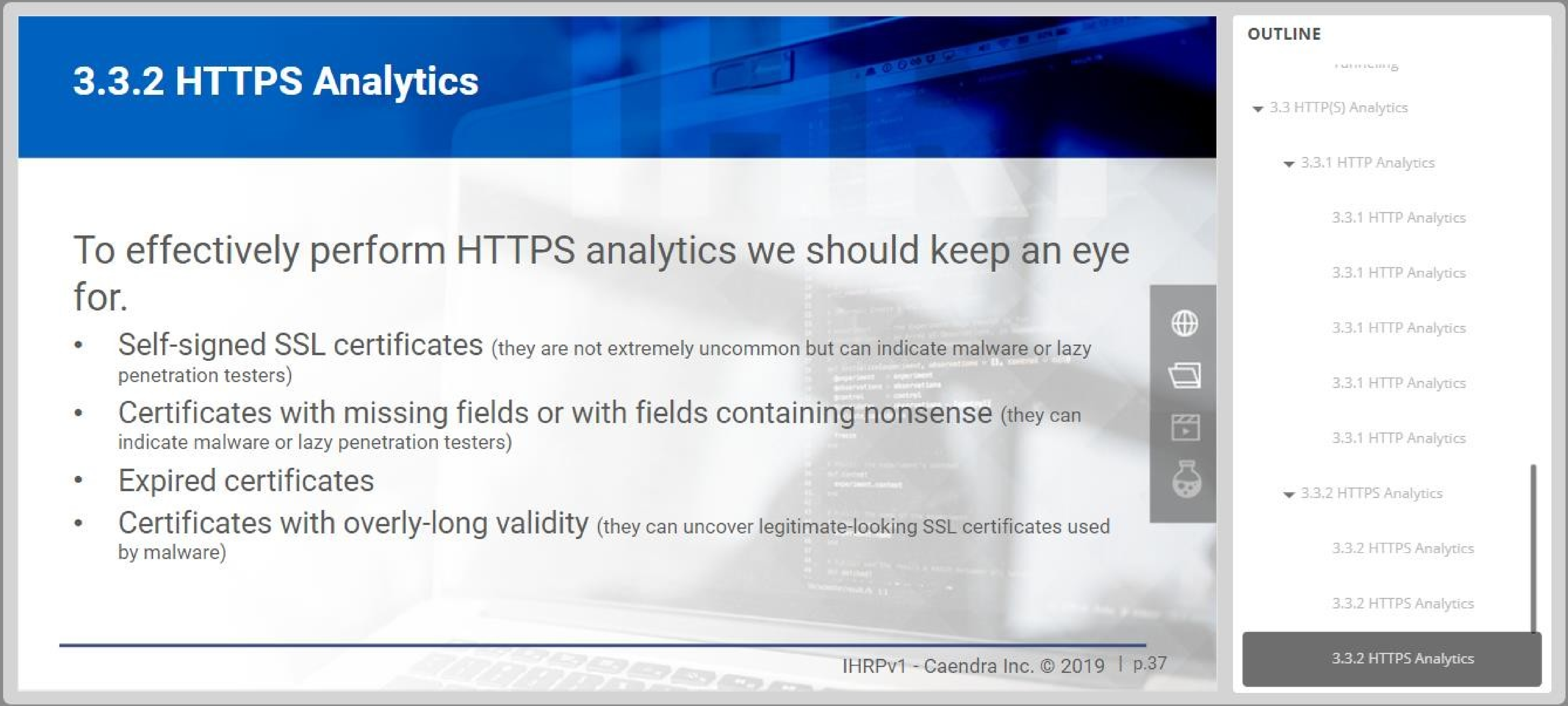
OUTLINE



3.3.2 HTTPS Analytics

The majority of the aforementioned sources log certificate-related fields such as *Timestamp*, *Source IP*, *Source Port*, *Destination IP*, *Destination Port*, *Key Algorithm*, *Key Length*, *Key Type*, *Not Valid After*, *Not Valid Before*, *Signing Algorithm*, *Subject*, *Version*, *Common Name*, *Organization*, *Organization Unit*, *Email*, *Issuer Info* etc.

OUTLINE

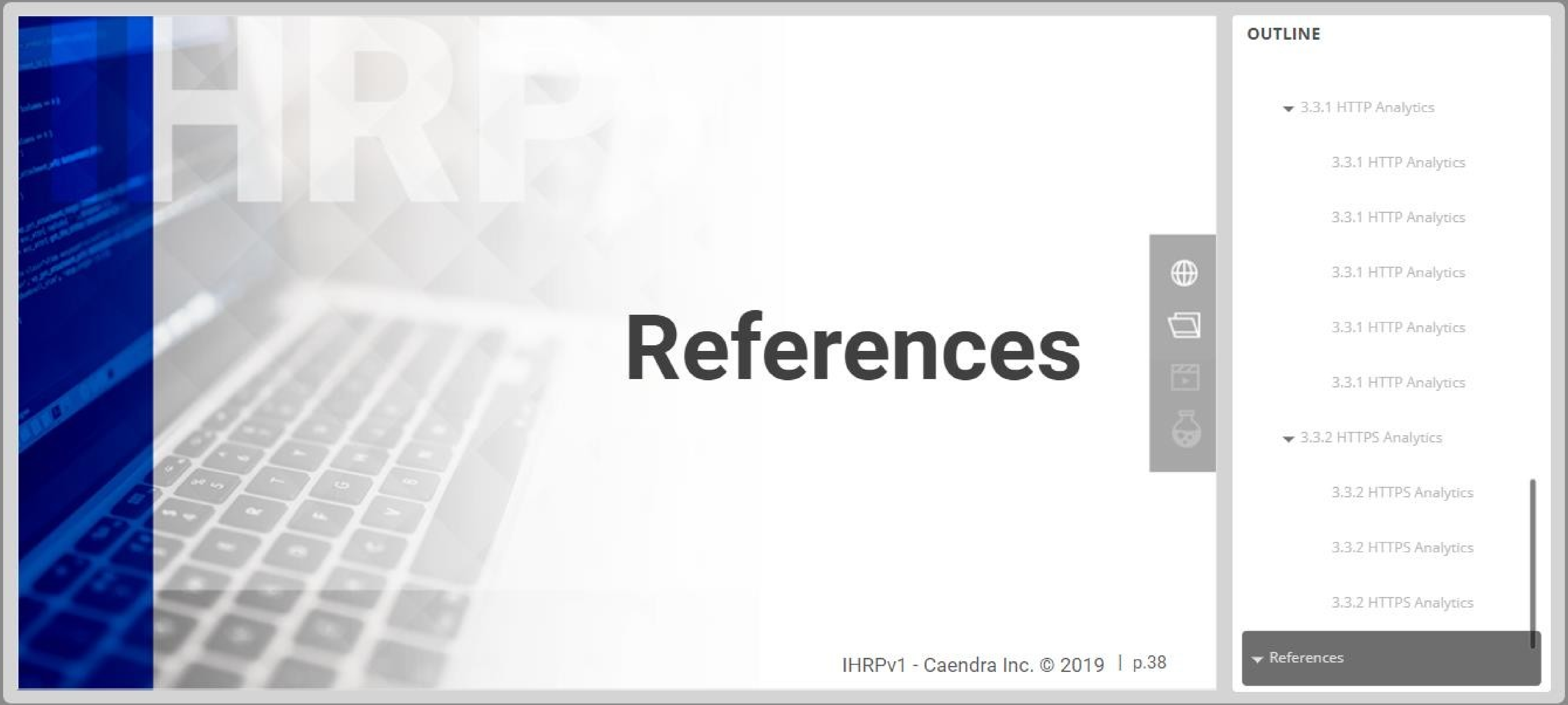


3.3.2 HTTPS Analytics

To effectively perform HTTPS analytics we should keep an eye for.

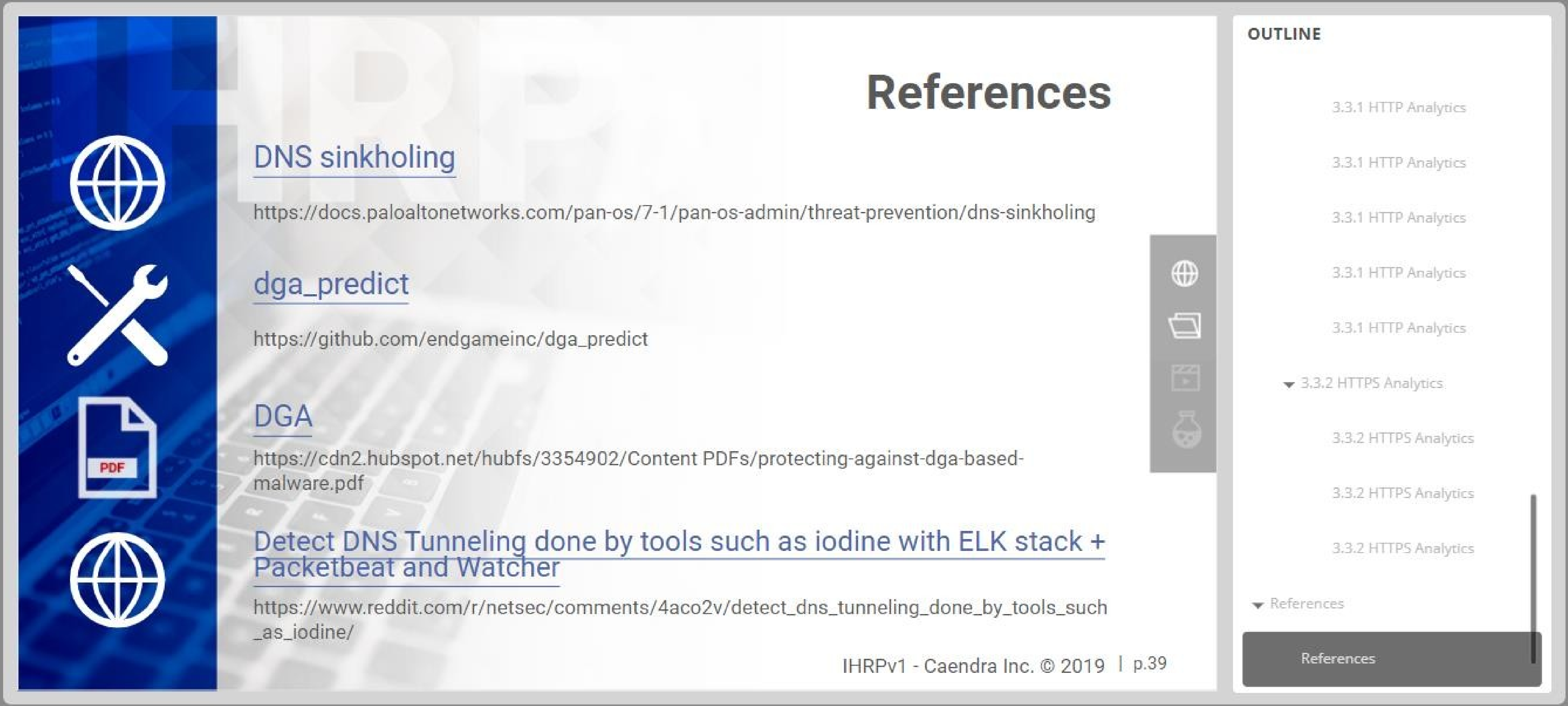
- Self-signed SSL certificates (they are not extremely uncommon but can indicate malware or lazy penetration testers)
- Certificates with missing fields or with fields containing nonsense (they can indicate malware or lazy penetration testers)
- Expired certificates
- Certificates with overly-long validity (they can uncover legitimate-looking SSL certificates used by malware)

OUTLINE



References

OUTLINE
3.3.1 HTTP Analytics
3.3.1 HTTP Analytics
3.3.1 HTTP Analytics
3.3.1 HTTP Analytics
3.3.1 HTTP Analytics
3.3.1 HTTP Analytics
3.3.2 HTTPS Analytics
3.3.2 HTTPS Analytics
3.3.2 HTTPS Analytics
3.3.2 HTTPS Analytics
References



References

DNS sinkholing

https://docs.paloaltonetworks.com/pan-os/7-1/pan-os-admin/threat-prevention/dns-sinkholing

dga_predict

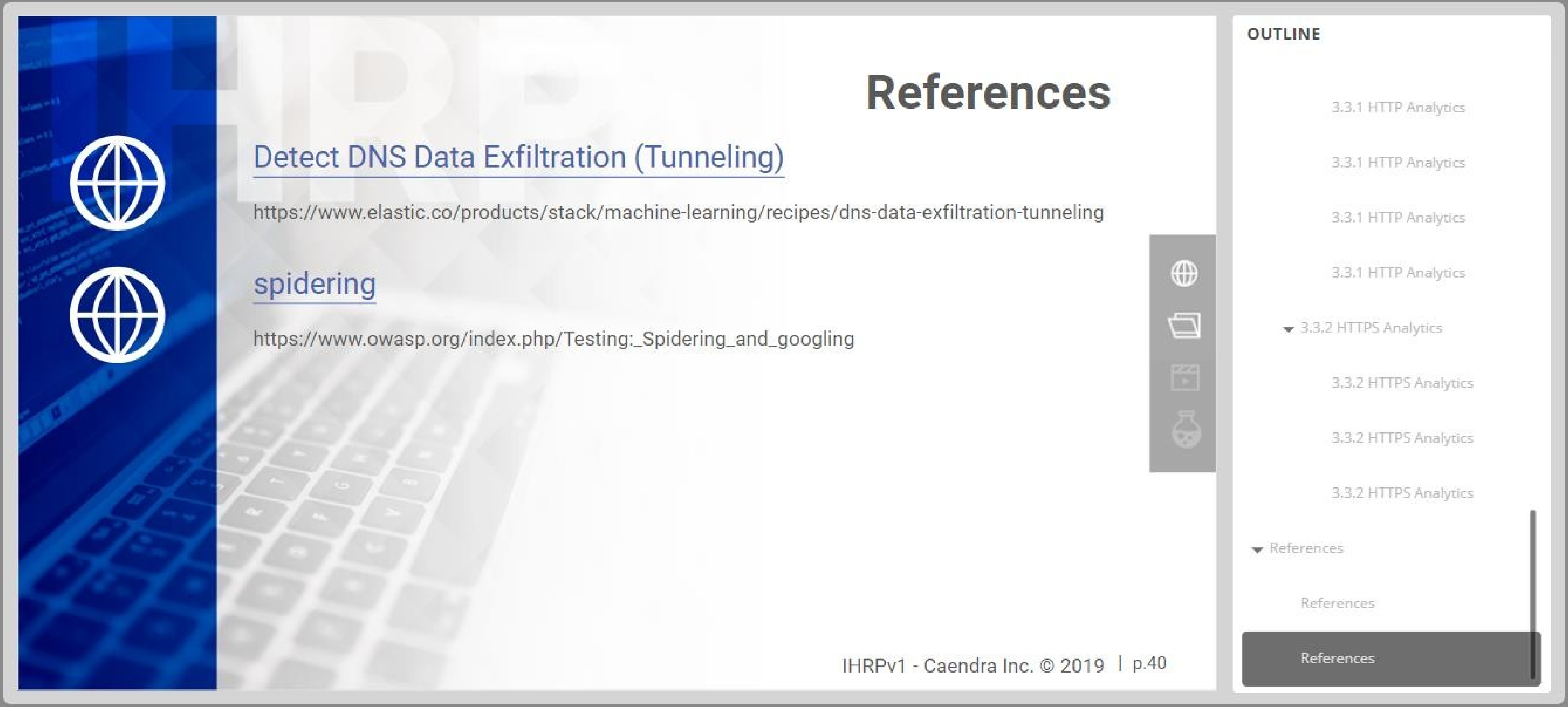
https://github.com/endgameinc/dga_predict

DGA

https://cdn2.hubspot.net/hubfs/3354902/Content PDFs/protecting-against-dga-based-malware.pdf

Detect DNS Tunneling done by tools such as iodine with ELK stack + Packetbeat and Watcher

https://www.reddit.com/r/netsec/comments/4aco2v/detect_dns_tunneling_done_by_tools_such_as_iodine/



References

Detect DNS Data Exfiltration (Tunneling)

https://www.elastic.co/products/stack/machine-learning/recipes/dns-data-exfiltration-tunneling

spidering

https://www.owasp.org/index.php/Testing:_Spidering_and_googling