Automatic Insecurity: Exploring Email Auto-configuration in the Wild

Shushang Wen[1], Yiming Zhang[2], Yuxiang Shen[1], Bingyu Li[3], Haixin Duan[2], Jingqiang Lin[1]

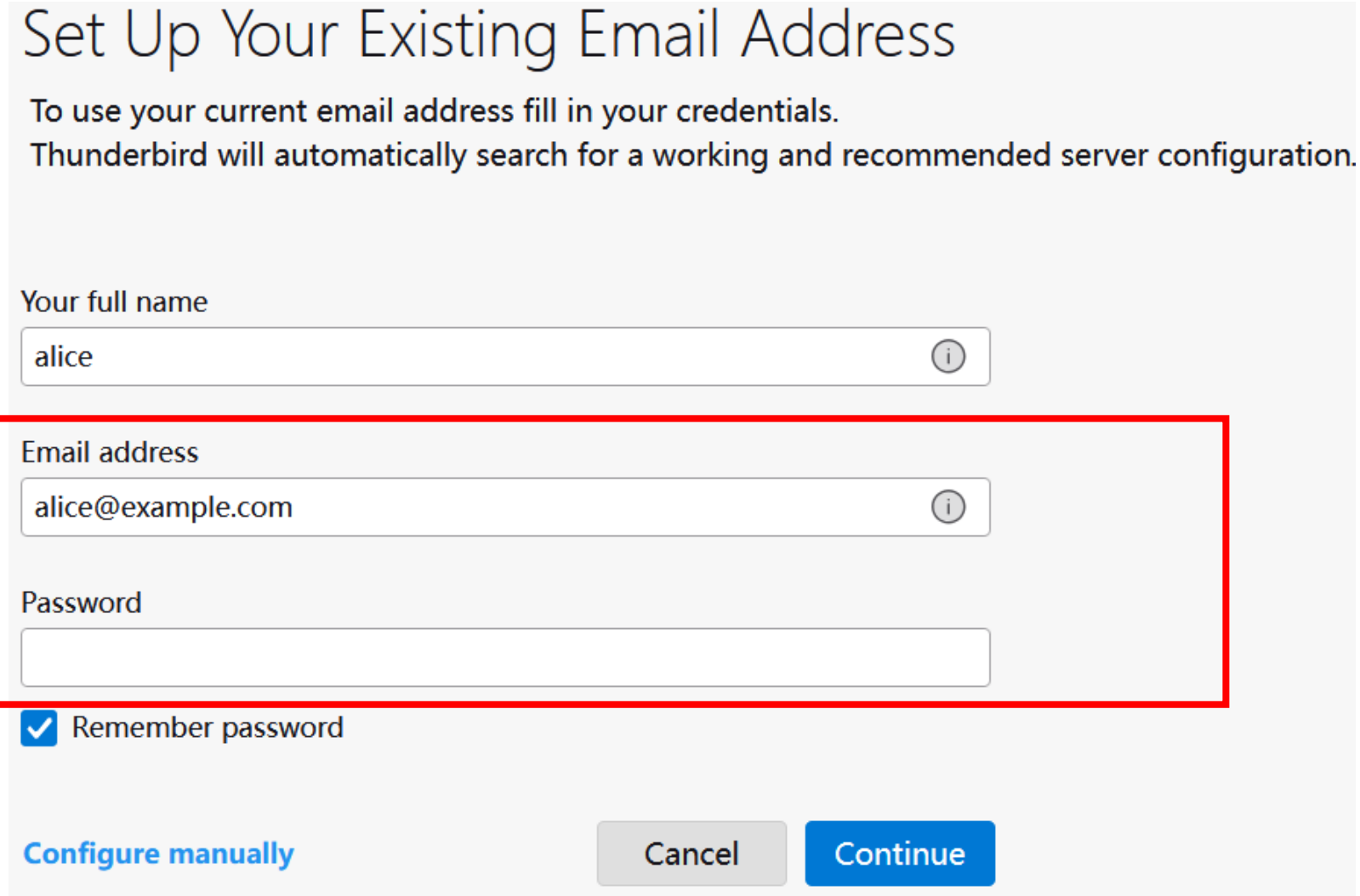
[1]University of Science and Technology of China, [2]Tsinghua University, [3]Beihang University

NDSS SYMPOSIUM/2025

Internet Society

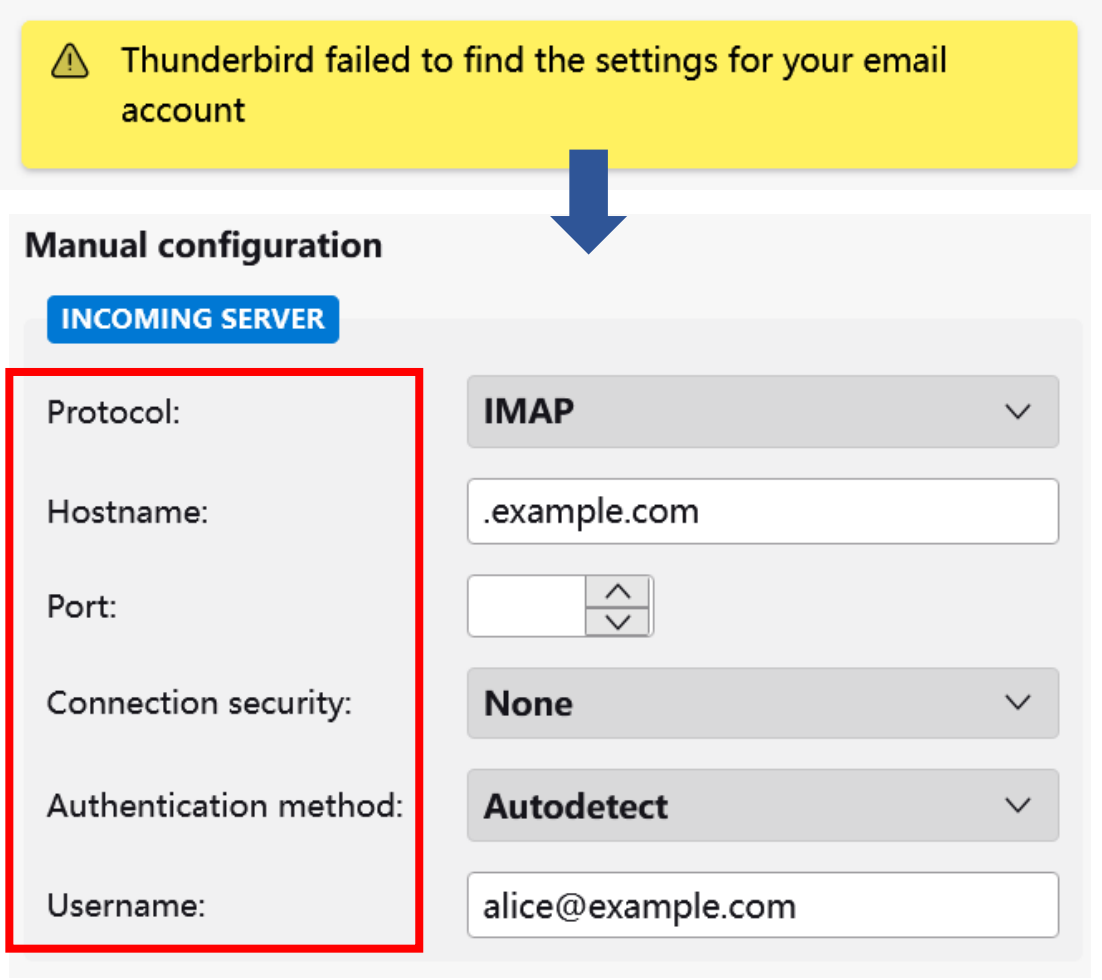
Email account setup: Thunderbird as an Example

- Typically, all we need to enter is the email address and password.



Email account setup: Thunderbird as an Example

- Typically, all we need to enter is the email address and password.
- However, if that fails, manual configuration is required.



Protocol: IMAP / POP3 / SMTP

Hostname: ?

Port: ?

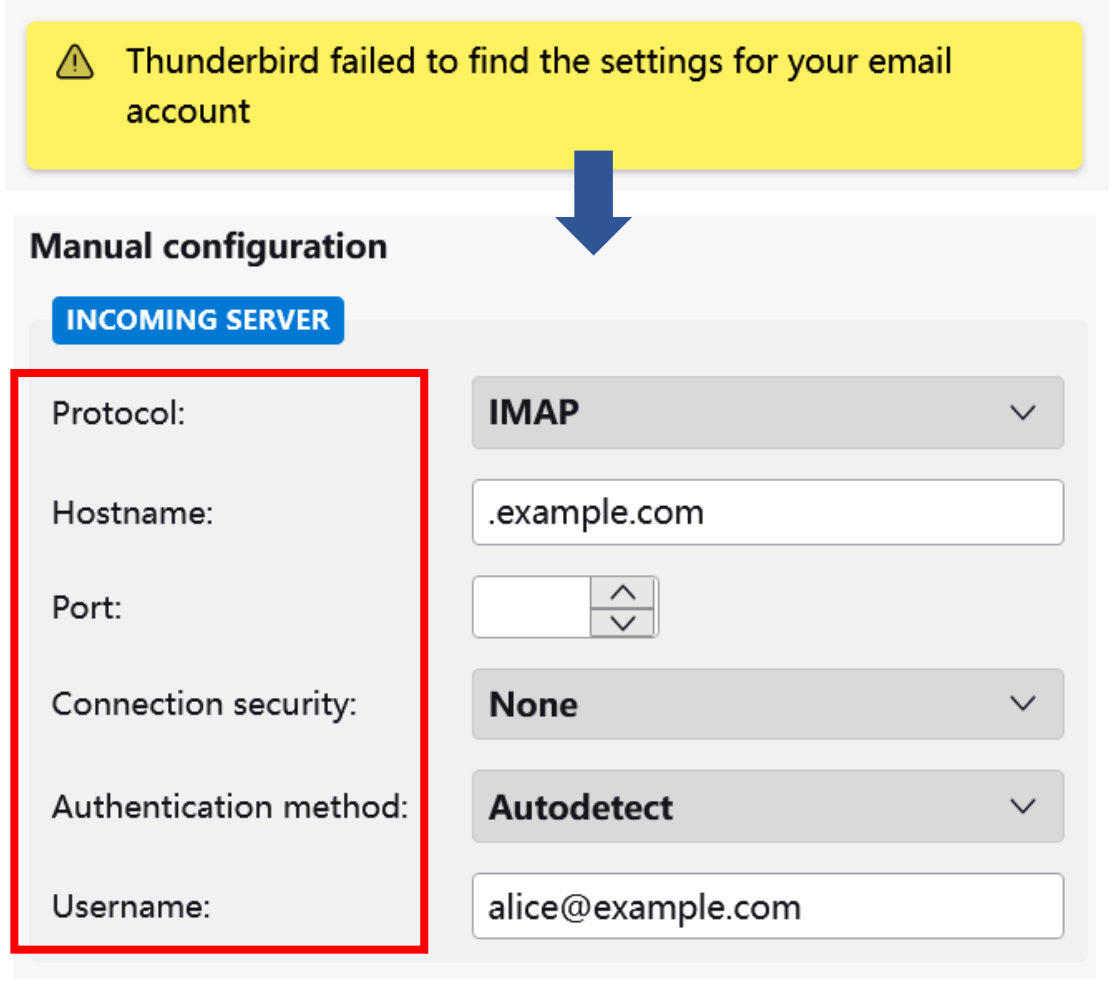
Connection security:

Autodetect / None / STARTTLS / TLS

...

Email account setup: Thunderbird as an Example

- Typically, all we need to enter is the email address and password.
- However, if that fails, manual configuration is required.



🤔 **How to fill in these configurations? Where to get it?**

Protocol: IMAP / POP3 / SMTP

Hostname: ?

Port: ?

Connection security:

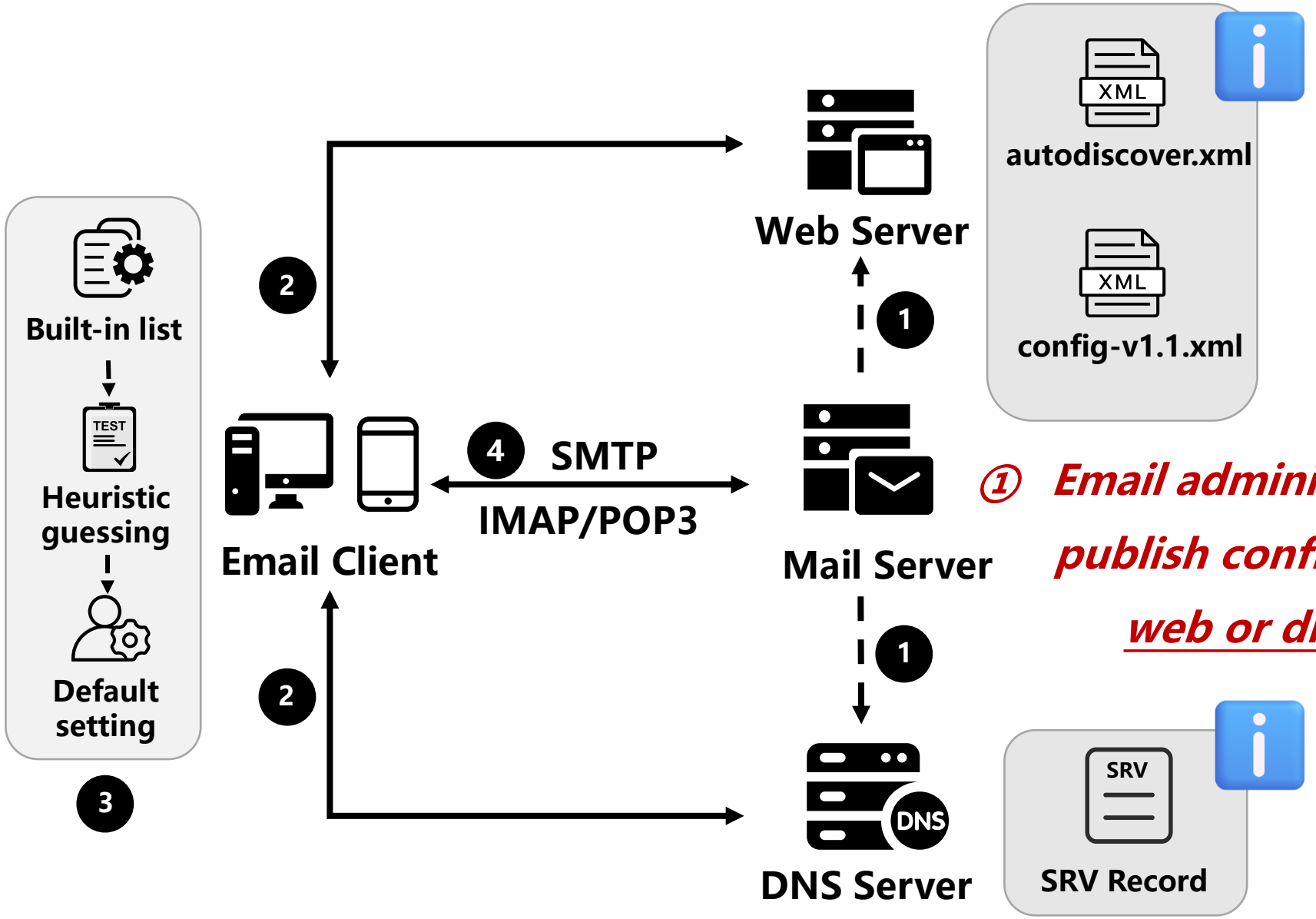
Autodetect / None / STARTTLS / TLS

…

Email auto-configuration

- **Define**: Automates the configuration process in email clients.
- We identified **6** common auto-configuration mechanisms involving both servers and clients.
  - **(To be) Standardized mechanisms:**
    - Autodiscover / Autoconfig / SRV service discovery **(RFC 6186 and 8314)**
  - **Besides, depend on clients:**
    - Built-in list / Heuristic guessing / Default settings.

The workflow of email auto-configuration



① *Email administrators pre-publish configurations in web or dns server.*

Example of autodiscover.xml & config-v1.1.xml

```xml
<Autodiscover xmlns="http://schemas.microsoft.com/exchange/a
    <Response xmlns="http://schemas.microsoft.com/exchange/a
        <Account>
            <AccountType>email</AccountType>
            <Action>settings</Action>
            <Protocol>
                <Type>IMAP</Type>
                <Server>imap.example.com</Server>
                <Port>993</Port>
                <SSL>on</SSL>
                <Encryption>SSL</Encryption>
                <SPA>off</SPA>
            </Protocol>
            <Protocol>
                <Type>SMTP</Type>
                <Server>smtp.example.com</Server>
                <Port>465</Port>
                <SSL>on</SSL>
                <Encryption>SSL</Encryption>
                <SPA>off</SPA>
            </Protocol>
        </Account>
    </Response>
</Autodiscover>
```

```xml
<clientConfig version="1.1">
    <emailProvider id="example.com">
        <domain>example.com</domain>
        <incomingServer type="imap">
            <hostname>imap.example.com</hostname>
            <port>993</port>
            <socketType>SSL</socketType>
            <authentication>password-cleartext</authentication>
            <username/>
        </incomingServer>
        <outgoingServer type="smtp">
            <hostname>smtp.example.com</hostname>
            <port>465</port>
            <socketType>SSL</socketType>
            <authentication>password-cleartext</authentication>
            <username/>
        </outgoingServer>
    </emailProvider>
</clientConfig>
```

[1] “[MS-OXDSCLI]: Autodiscover HTTP Service Protocol,” https://msopenspecs.azureedge.net/files/MS-OXDSCLI/%5bMS-OXDSCLI%5d-210817.pdf
[2] “Mail Autoconfig,” Internet Engineering Task Force, Internet-Draft draft-bucksch-autoconfig-00, https://datatracker.ietf.org/doc/draft-bucksch-autoconfig/00/

Example of autodiscover.xml & config-v1.1.xml

```xml
<Autodiscover xmlns="http://schemas.microsoft.com/exchange/a
    <Response xmlns="http://schemas.microsoft.com/exchange/a
        <Account>
            <AccountType>email</AccountType>
            <Action>settings</Action>
            <Protocol>
                <Type>IMAP</Type>
                <Server>imap.example.com</Server>
                <Port>993</Port>
                <SSL>on</SSL>
                <Encryption>SSL</Encryption>

                <Server>smtp.example.com</Server>
                <Port>465</Port>
                <SSL>on</SSL>
                <Encryption>SSL</Encryption>
                <SPA>off</SPA>
            </Protocol>
        </Account>
    </Response>
</Autodiscover>
```

```xml
<clientConfig version="1.1">
    <emailProvider id="example.com">
        <domain>example.com</domain>
        <incomingServer type="imap">
            <hostname>imap.example.com</hostname>
            <port>993</port>
            <socketType>SSL</socketType>
            <authentication>password-cleartext</authentication>
            <username/>
            <port>465</port>
            <socketType>SSL</socketType>
            <authentication>password-cleartext</authentication>
            <username/>
        </outgoingServer>
    </emailProvider>
</clientConfig>
```

<Server> [1] or <hostname> [2] determine the destination to which the client connects.

[1] "[MS-OXDSCLI]: Autodiscover HTTP Service Protocol," https://msopenspecs.azureedge.net/files/MS-OXDSCLI/%5bMS-OXDSCLI%5d-210817.pdf

[2] "Mail Autoconfig," Internet Engineering Task Force, Internet-Draft draft-bucksch-autoconfig-00, https://datatracker.ietf.org/doc/draft-bucksch-autoconfig/00/

Example of autodiscover.xml & config-v1.1.xml

```xml
<Autodiscover xmlns="http://schemas.microsoft.com/exchange/a
    <Response xmlns="http://schemas.microsoft.com/exchange/a
        <Account>
            <AccountType>email</AccountType>
            <Action>settings</Action>
            <Protocol>
                <Type>IMAP</Type>
                <Server>imap.example.com</Server>
                <Port>993</Port>
                <SSL>on</SSL>
                <Encryption>SSL</Encryption>
                <SPA>off</SPA>
            </Protocol>
            <Protocol>
                <Type>SMTP</Type>
                <Server>smtp.example.com</Server>
                <Port>587</Port>
                <SSL>on</SSL>
                <Encryption>SSL</Encryption>
                <SPA>off</SPA>
            </Protocol>
        </Account>
    </Response>
</Autodiscover>
```

```xml
<clientConfig version="1.1">
    <emailProvider id="example.com">
        <domain>example.com</domain>
        <incomingServer type="imap">
            <hostname>imap.example.com</hostname>
            <port>993</port>
            <socketType>SSL</socketType>
            <authentication>password-cleartext</authentication>
            <username/>
        </incomingServer>
        <outgoingServer type="smtp">
            <hostname>smtp.example.com</hostname>
            <port>587</port>
            <socketType>SSL</socketType>
            <authentication>password-cleartext</authentication>
            <username/>
        </outgoingServer>
    </emailProvider>
</clientConfig>
```

<SSL>, <Encryption> [1], or <socketType> [2] determine whether an encrypted connection (*client* ↔ *mail server*) is required.

[1] "[MS-OXDSCLI]: Autodiscover HTTP Service Protocol," https://msopenspecs.azureedge.net/files/MS-OXDSCLI/%5bMS-OXDSCLI%5d-210817.pdf
[2] "Mail Autoconfig," Internet Engineering Task Force, Internet-Draft draft-bucksch-autoconfig-00, https://datatracker.ietf.org/doc/draft-bucksch-autoconfig/00/

Definition of SRV service discovery

- **RFC 6186 [1] & RFC 8314 [2]**
  - A method for locating email submission and access services, specifying the hostname, port, and connection type (i.e., whether TLS is supported).
  - Lower ***Priority*** values are preferred.

<service labels>.<example.com> SRV <Priority> <Weight> <Port> <Hostname>

An example of service records

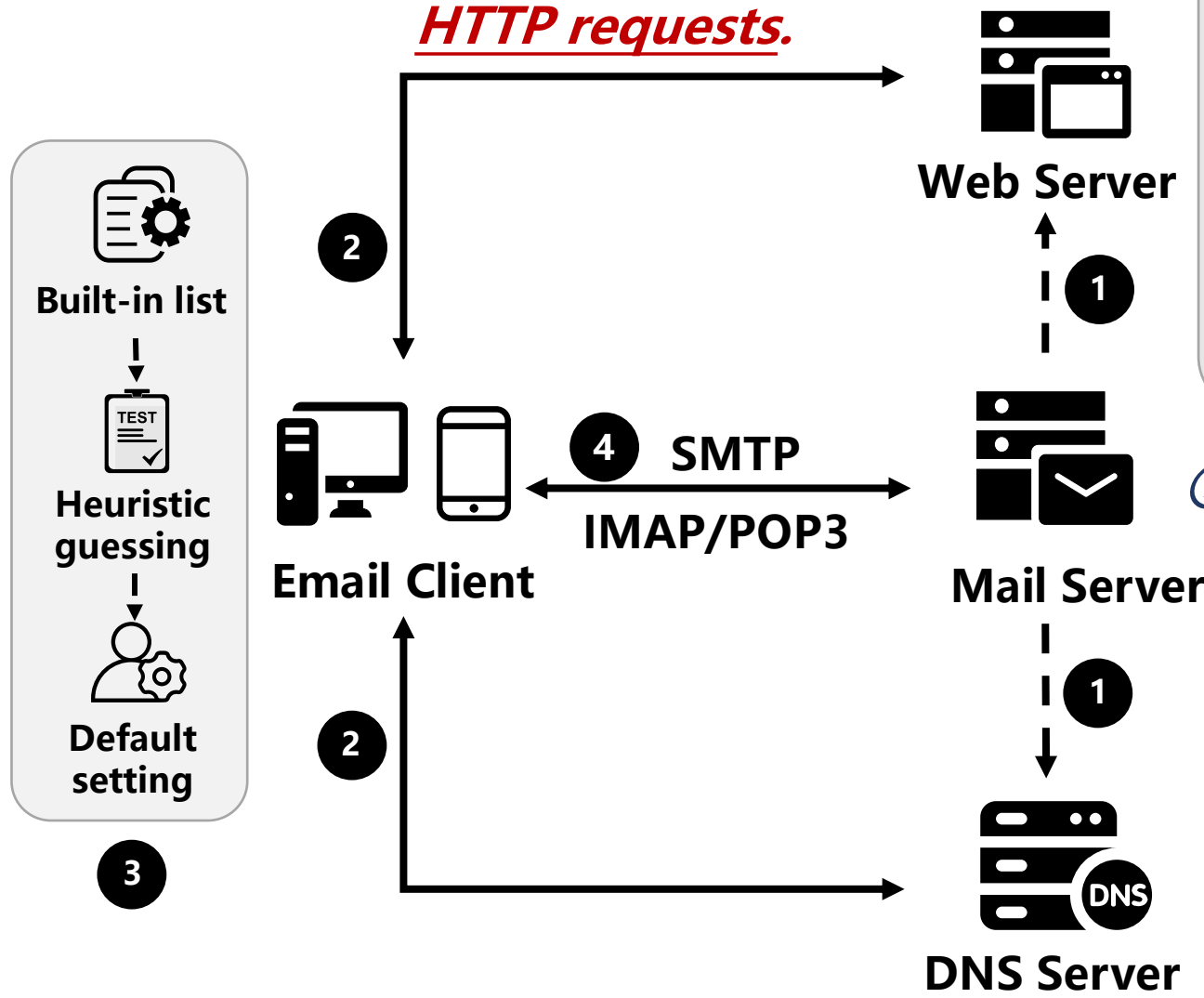
```
_imap._tcp.example.com      SRV  0   0   143   .
_imaps._tcp.example.com     SRV  0   1   993   imap.example.com.
_pop3._tcp .example.com     SRV  0   0   110   .
_pop3s._tcp .example.com    SRV  10  1   995   pop3.example.com.
```

✓ **Preferred**

[1] "Use of SRV records for locating email submission/access services," RFC, vol. 6186, pp. 1–9, 2011. [Online]. Available: https://doi.org/10.17487/RFC6186
[2] "Cleartext considered obsolete: Use of transport layer security (TLS) for email submission and access," RFC, vol. 8314, pp. 1–26, Jan. 2018. [Online]. Available: https://doi.org/10.17487/RFC8314

The workflow of email auto-configuration

② *Client retrieves configurations through HTTP requests.*



③ *Client refers to other mechanisms.*

① *Email administrators pre-publish configurations in web or dns server.*

② *Client retrieves configurations through DNS requests.*

Configuration HTTP requests - Autoconfig

- **Autoconfig requests [1]**

e.g., alice@example.com

HTTP GET request

http(s)://autoconfig.**example.com**/mail/config-v1.1.xml?emailaddress=**alice@example.com**
https://**example.com**/.well-known/autoconfig/mail/config-v1.1.xml?emailaddress=**alice@example.com**
…

HTTP GET request based on MX hostname

example.com. MX 0 **mx.backoff.target.com.**
https://autoconfig.**backoff.target.com**/mail/config-v1.1.xml?emailaddress=**alice@example.com**
https://autoconfig.**target.com**/mail/config-v1.1.xml?emailaddress=**alice@example.com**
…

Local import

%USER_CONFIGURATION_DIR%/isp/**example.com**.xml

[1] "Mail Autoconfig," Internet-Draft draft-bucksch-autoconfig-00, https://datatracker.ietf.org/doc/draft-bucksch-autoconfig/00/

Configuration HTTP requests - Autodiscover, similarly

- **Autodiscover requests [1]**

e.g., alice@example.com

HTTP POST request

http://**example.com**/autodiscover/autodiscover.xml
https://autodiscover.**example.com**/autodiscover/autodiscover.xml

SRV for Autodiscover server, then POST

_autodiscover._tcp.**example.com.** SRV 0 0 443 **target.com.**
https://**target.com**/autodiscover/autodiscover.xml

HTTP GET for initial, POST for redirection

http://autodiscover.**example.com**/autodiscover/autodiscover.xml

[1] "[MS-OXDISCO]: Autodiscover HTTP Service Protocol," https://msopenspecs.azureedge.net/files/MS-OXDISCO/%5bMS-OXDISCO%5d-210817.pdf

Configuration DNS requests - SRV service labels

- **SRV requests**
  - RFC 6186 [1] and 8314 [2] define 6 **service labels** based on the combination of the protocol and the security mechanism.

e.g., alice@example.com

Plaintext or STARTTLS ⬇

_submission._tcp.**example.com**
_imap._tcp.**example.com**
_pop3._tcp.**example.com**

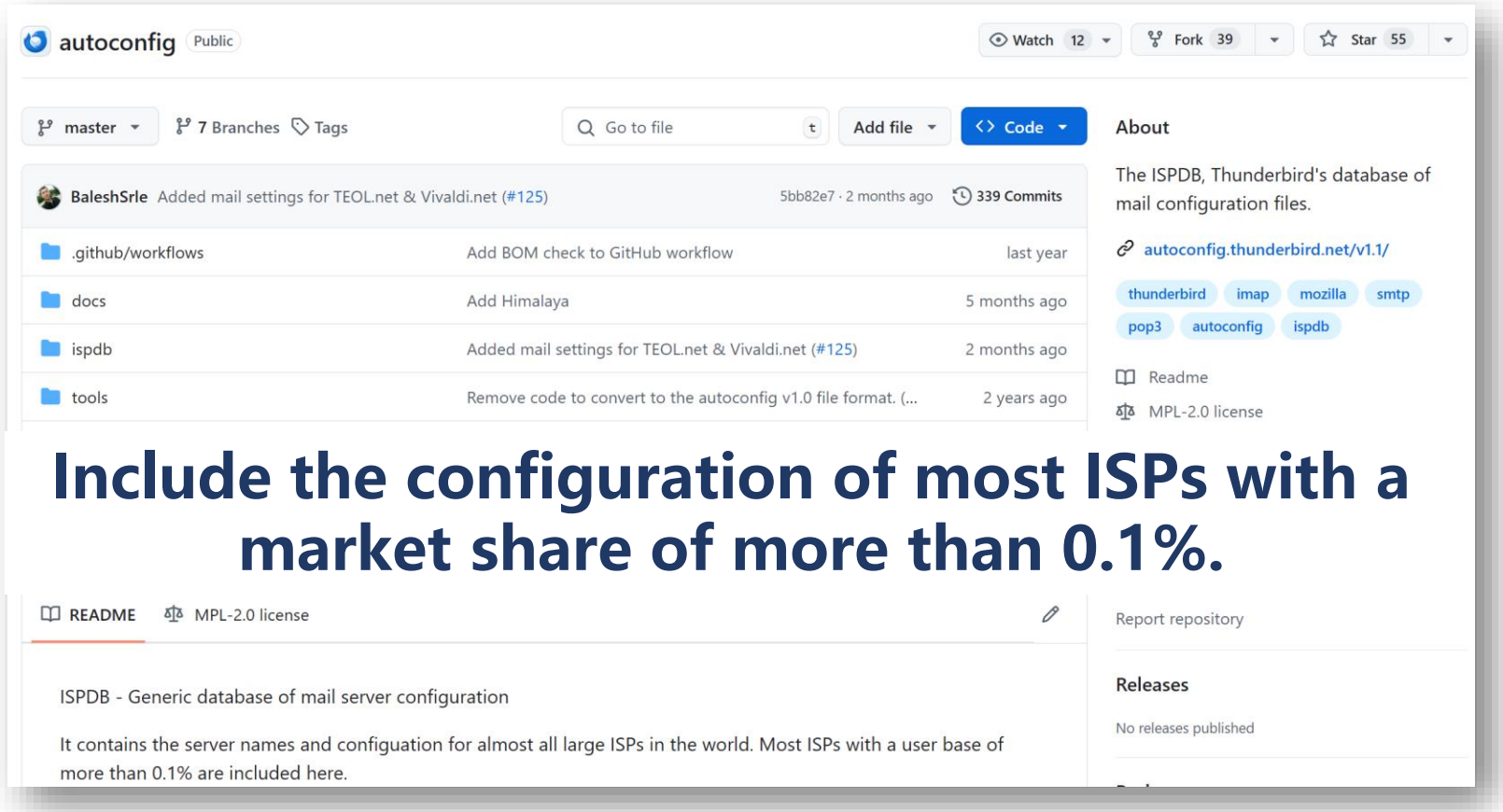
⬇ **implicit TLS**

_submission**s**._tcp.**example.com**
_imap**s**._tcp.**example.com**
_pop3**s**._tcp.**example.com**

[1] "Use of SRV records for locating email submission/access services," RFC, vol. 6186, pp. 1–9, 2011. [Online]. Available: https://doi.org/10.17487/RFC6186
[2] "Cleartext considered obsolete: Use of transport layer security (TLS) for email submission and access," RFC, vol. 8314, pp. 1–26, Jan. 2018. [Online]. Available: https://doi.org/10.17487/RFC8314

Other mechanisms – E.g., built-in list

- **Built-in configuration information for popular mail providers.**
  - For example, ISPDB [1], maintained by Thunderbird, is widely used on the client side (14/29 in our tests).



Include the configuration of most ISPs with a market share of more than 0.1%.

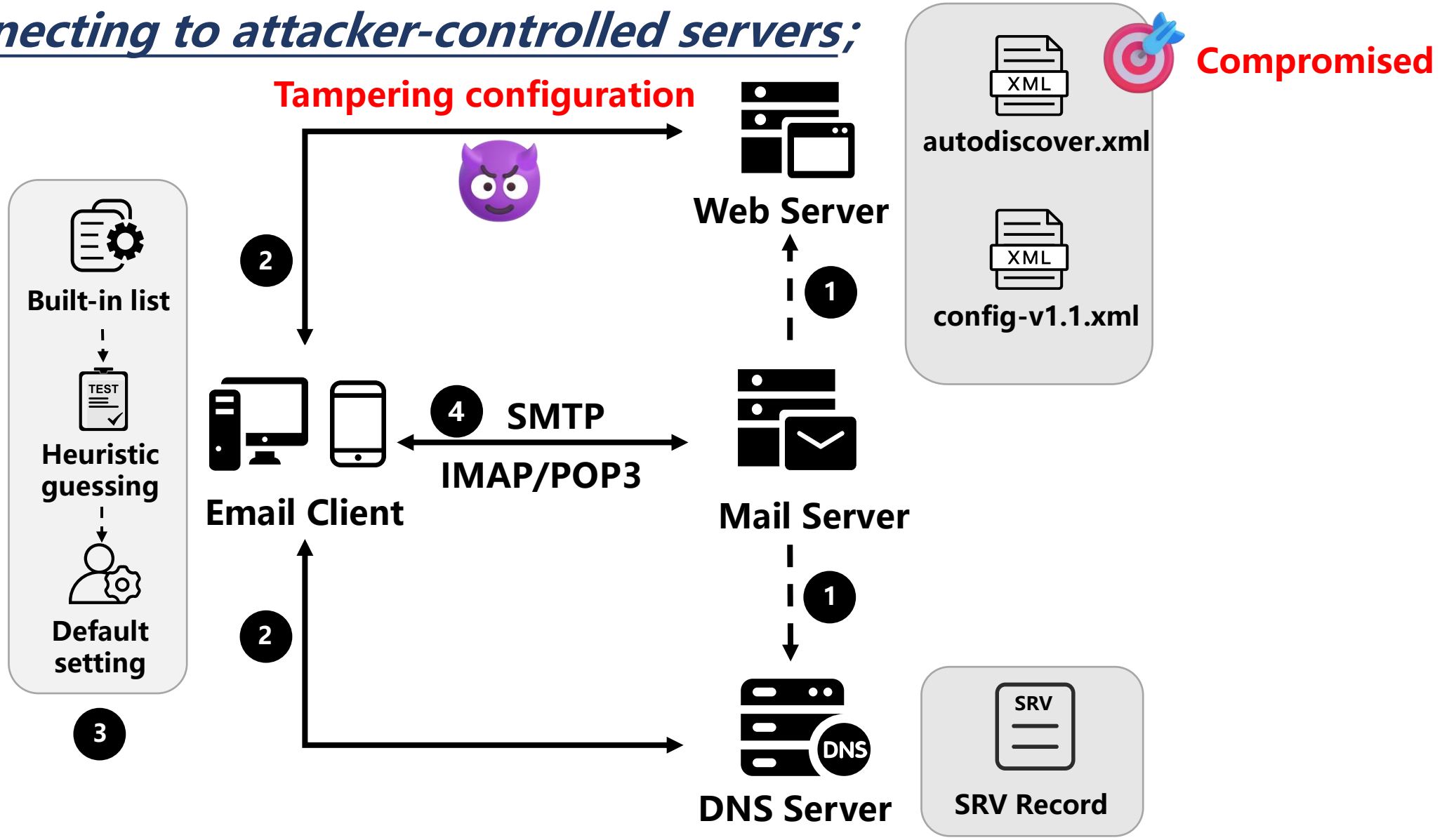
[1] "Ispdb - generic database of mail server configuration," https://github.com/thunderbird/autoconfig/tree/master/ispdb

Research questions

- **What security threats exist in email auto-configuration?**
  - Configuration information transmitted securely?
  - Server-provided configuration instructing client to establish secure connections?
- **How extensive is their impact on email services?**
  - Misconfigured servers? Flawed client implementation?

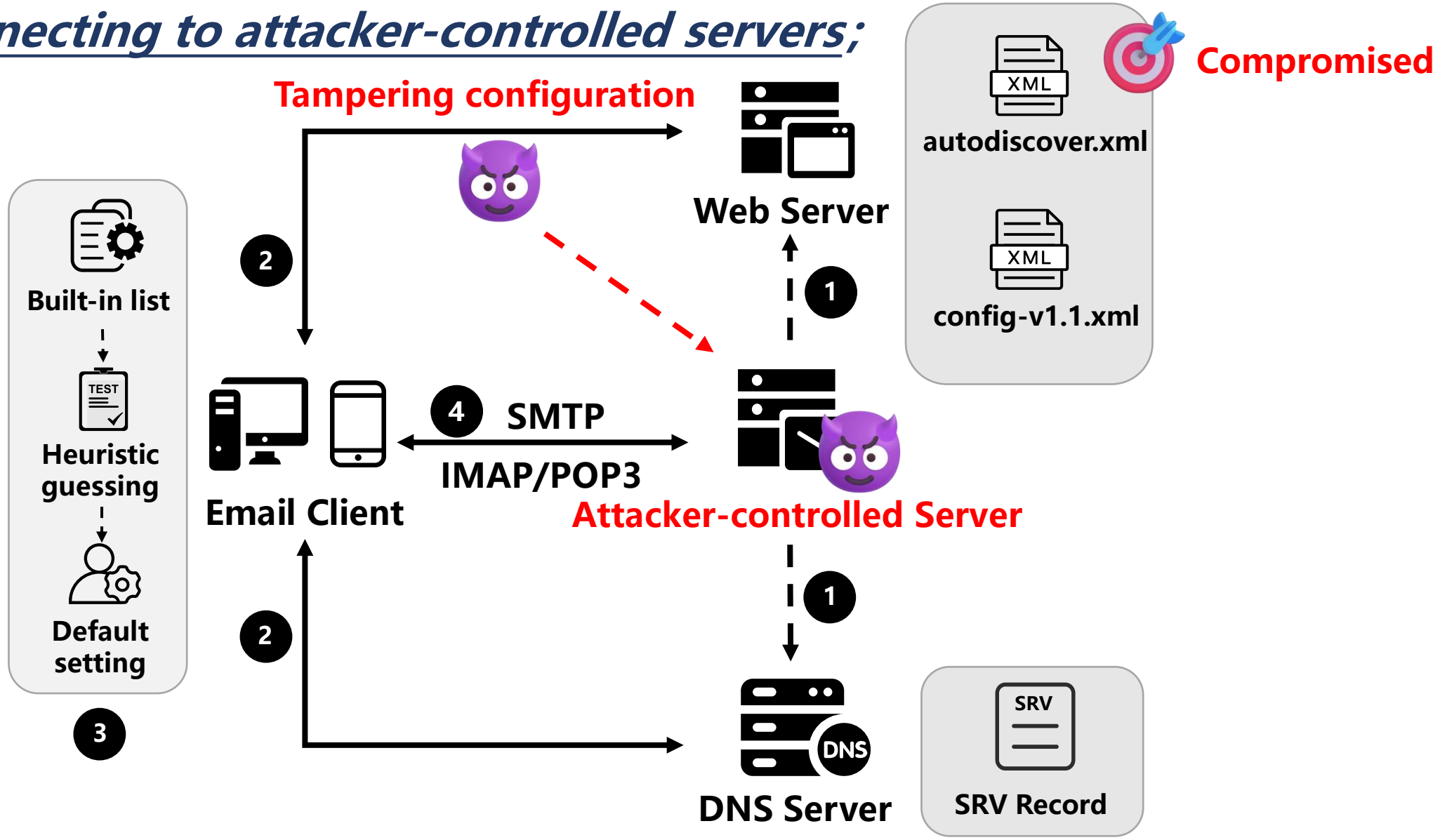
Attack goal

- *Type-I: Connecting to attacker-controlled servers;*



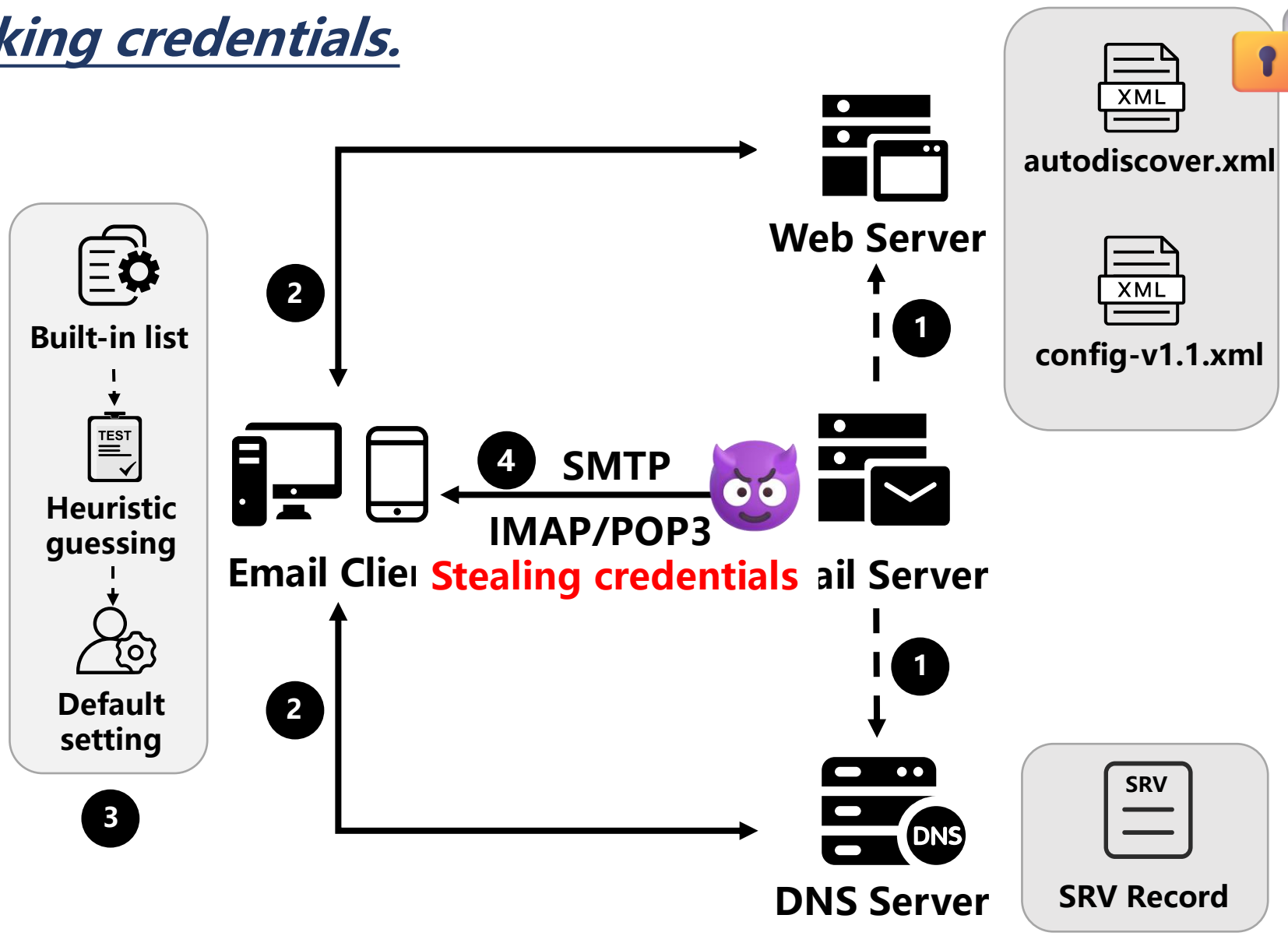
Attack goal

- *Type-I: Connecting to attacker-controlled servers;*



Attack goal

- ***Type-II: Leaking credentials.***



* We do not consider DNS resolve-related threats here as it is applied to all network application relying on domains.

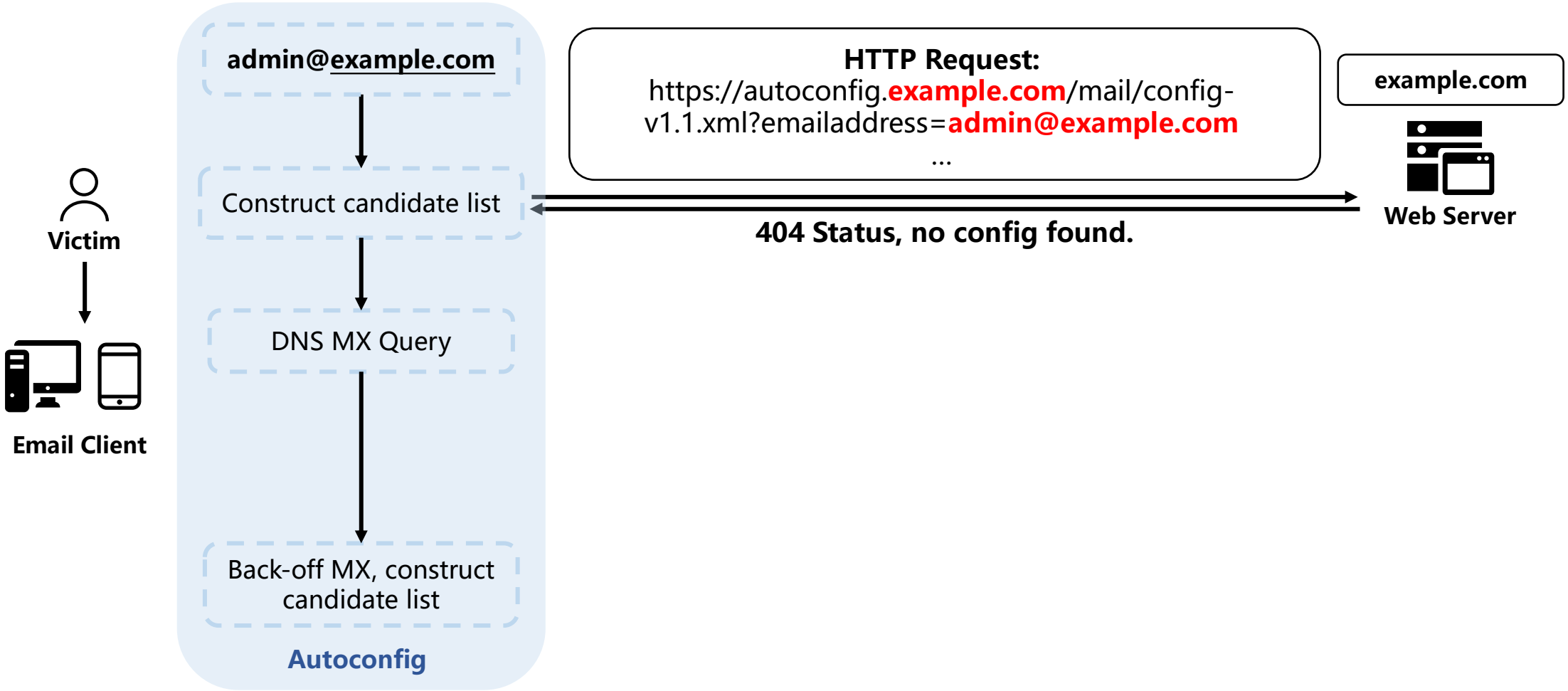
Attack case (A2.1): Inadequate eTLD validation

- Inadequate eTLD validation when extracting the domain part of an email address to construct an Autoconfig request.

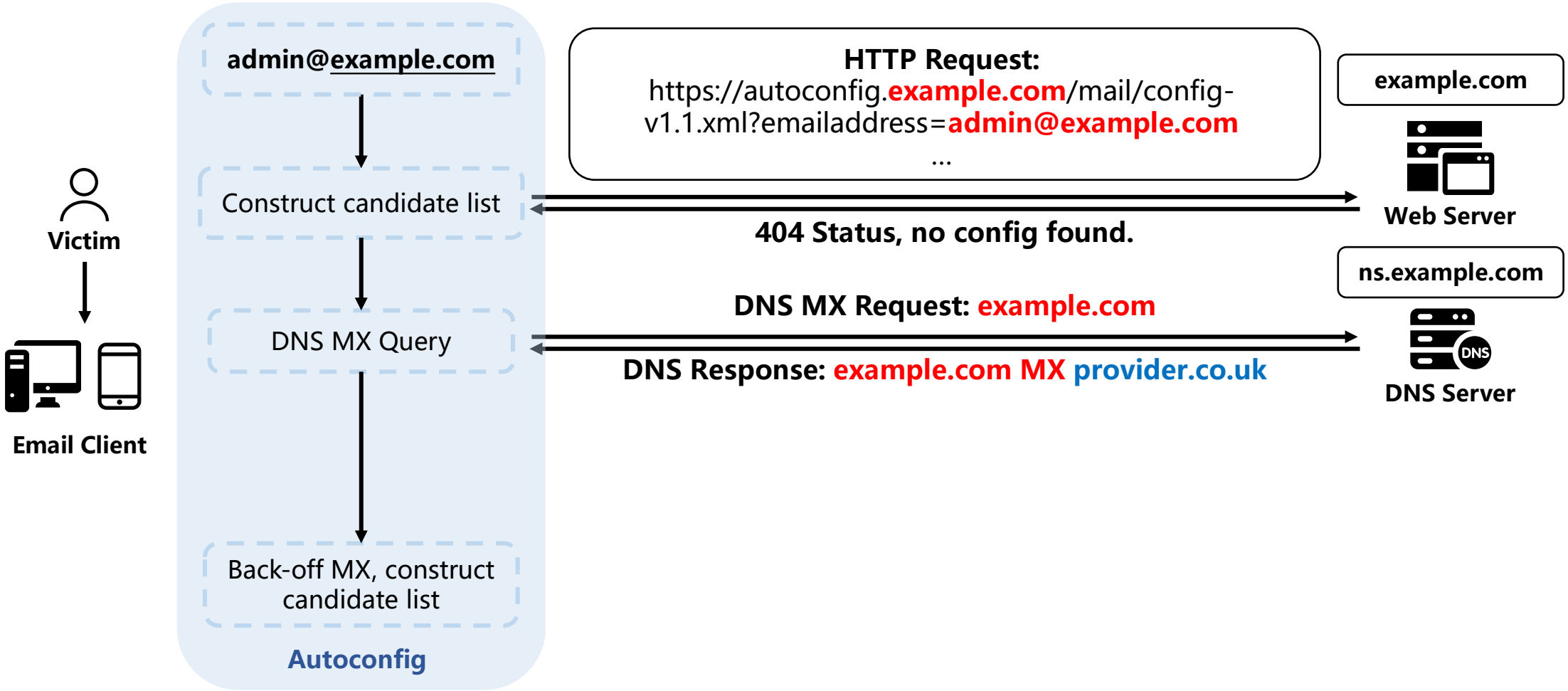
Type-I: Connecting to attacker-controlled servers;

- For ***admin@example.com***, the following conditions are required:
  - *example.com* has not deployed Autoconfig.
  - MX hostname for example.com is an eTLD+1 (e.g., *provider.co.uk*).
  - Attacker can register *autoconfig.co.uk*.

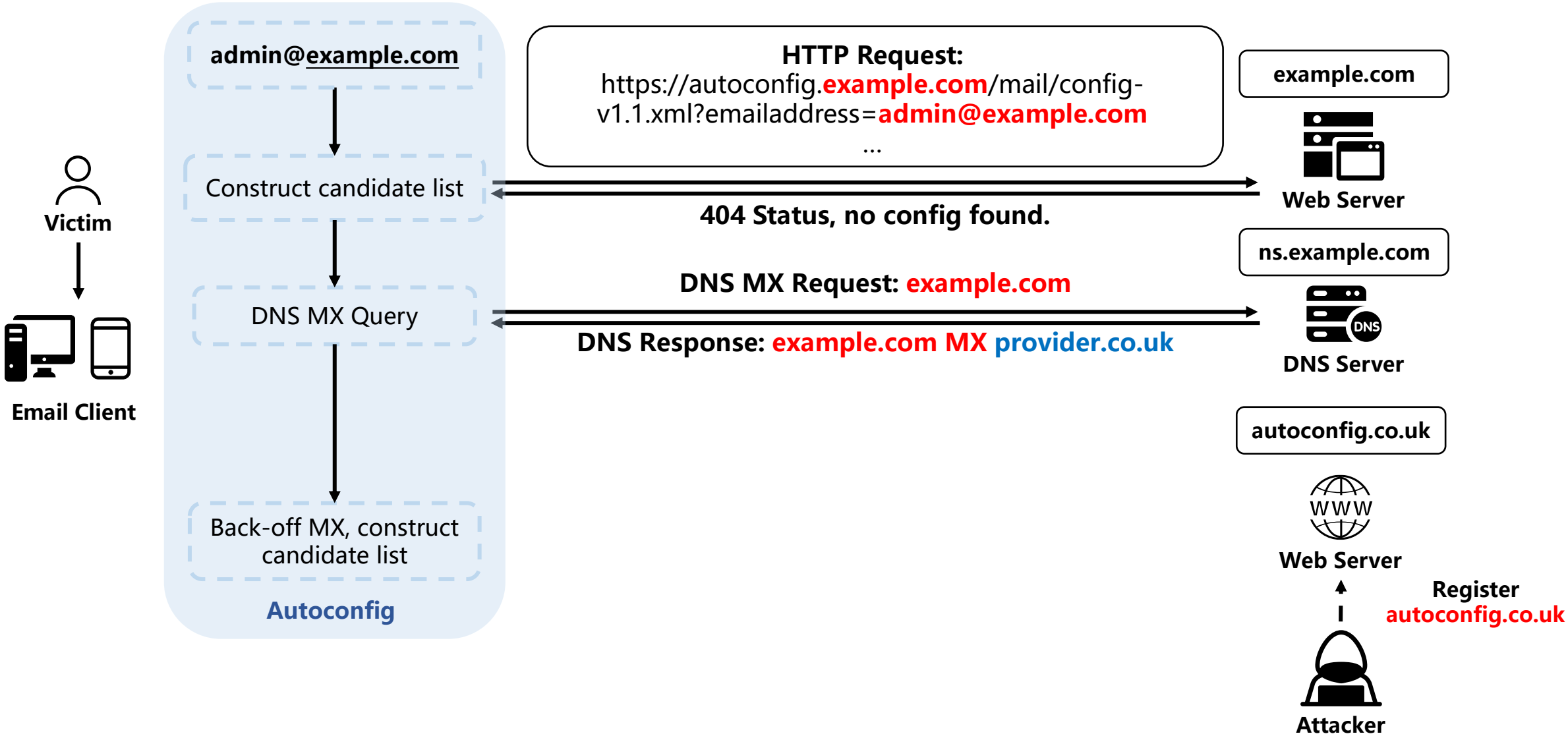
Attack case (A2.1): Inadequate eTLD validation



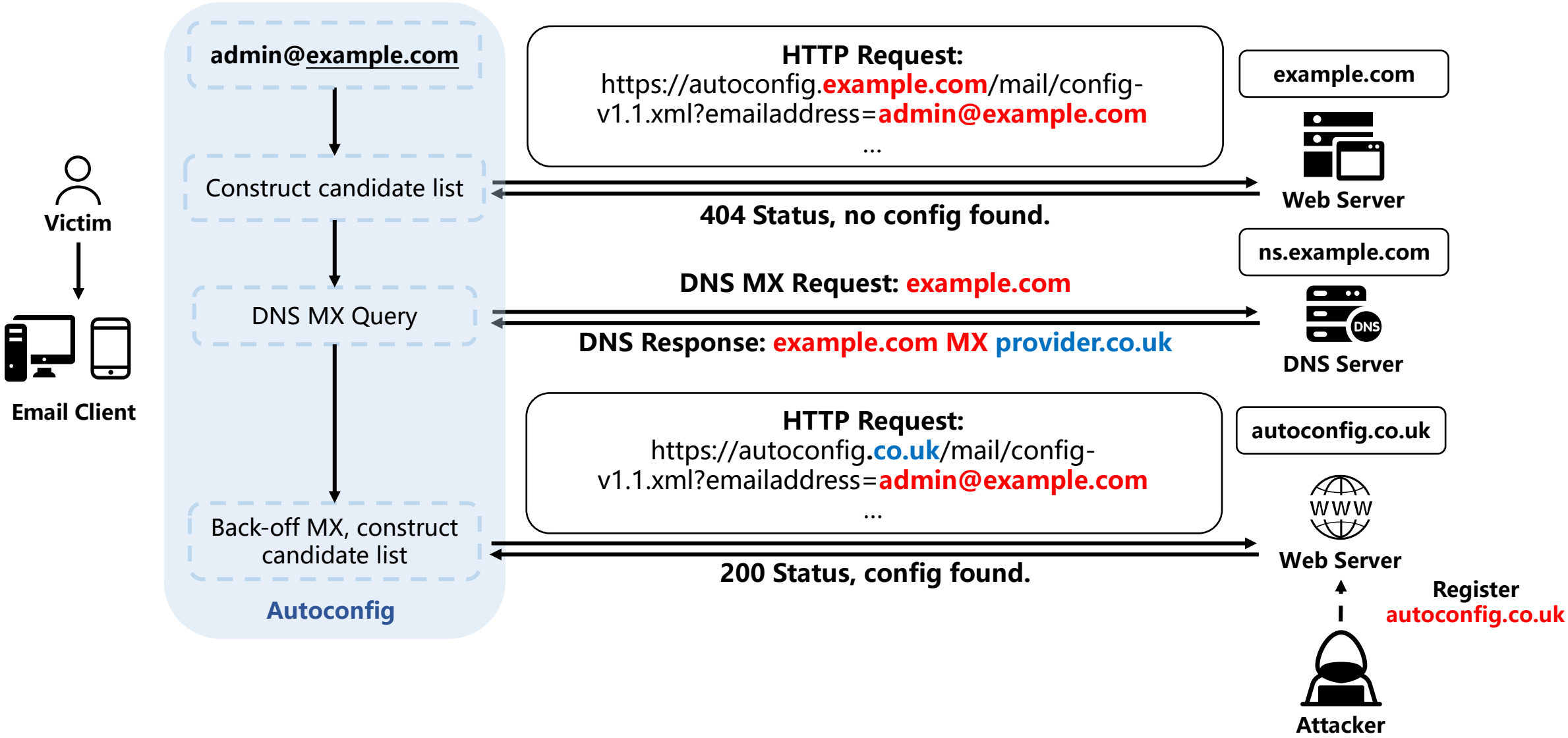
Attack case (A2.1): Inadequate eTLD validation



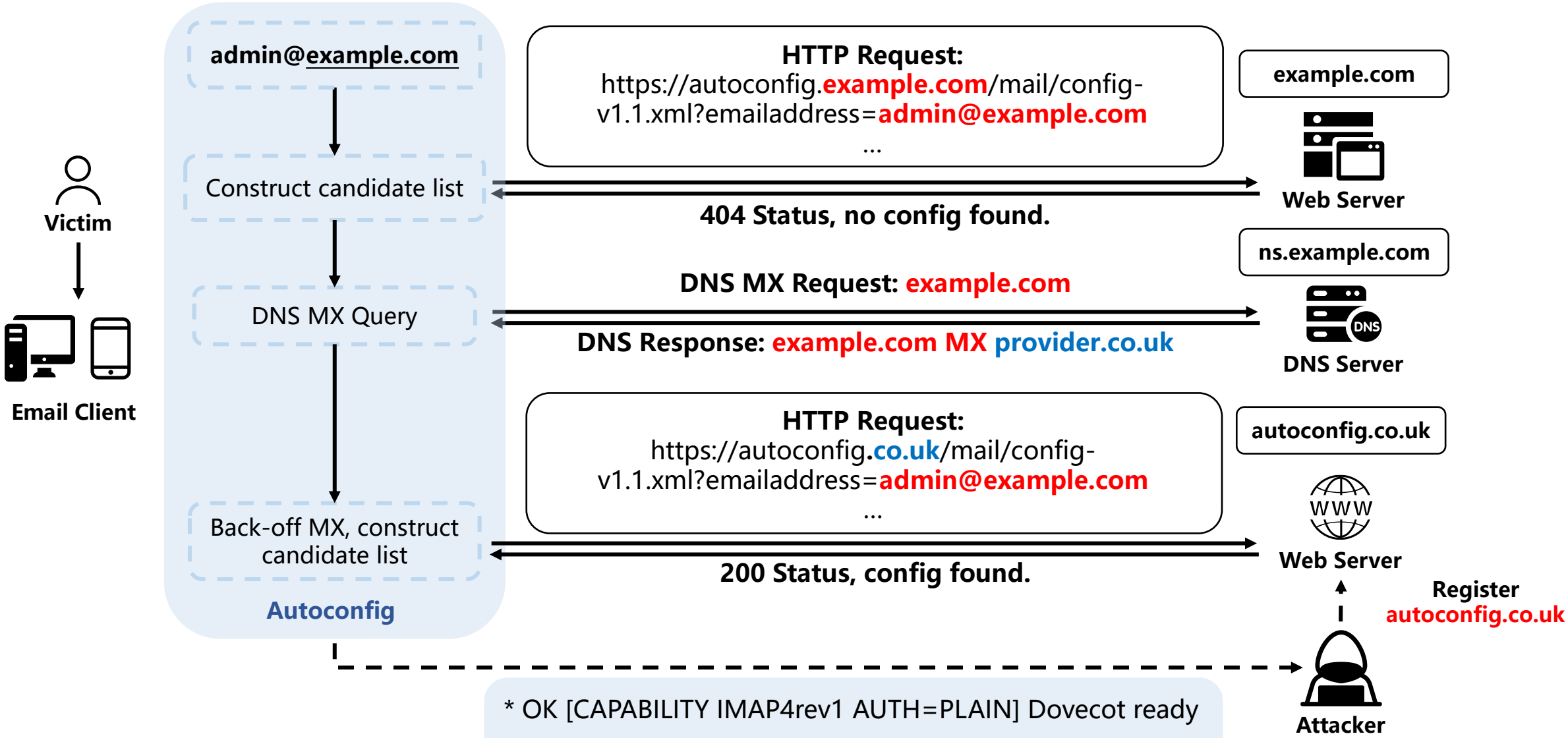
Attack case (A2.1): Inadequate eTLD validation



Attack case (A2.1): Inadequate eTLD validation



Attack case (A2.1): Inadequate eTLD validation



Connecting to attacker-controlled server

Attack case (A10.1): Inconsistent connection type

- The server administrator published both config-v1.1.xml and autodiscover.xml at the same time, but the configurations contained were inconsistent.

😈 ***Type-II: Leaking credentials;***

- For example,

```
1   <clientConfig version="1.1">
2       <emailProvider id="example.com">
3           <domain>example.com</domain>
4           <incomingServer type="imap">
5               <hostname>imap.example.com</hos
6               <port>993</port>
7               <socketType>SSL</socketType>
8               <authentication>password-cleart
9               <username/>
10          </incomingServer>
```

config-v1.1.xml

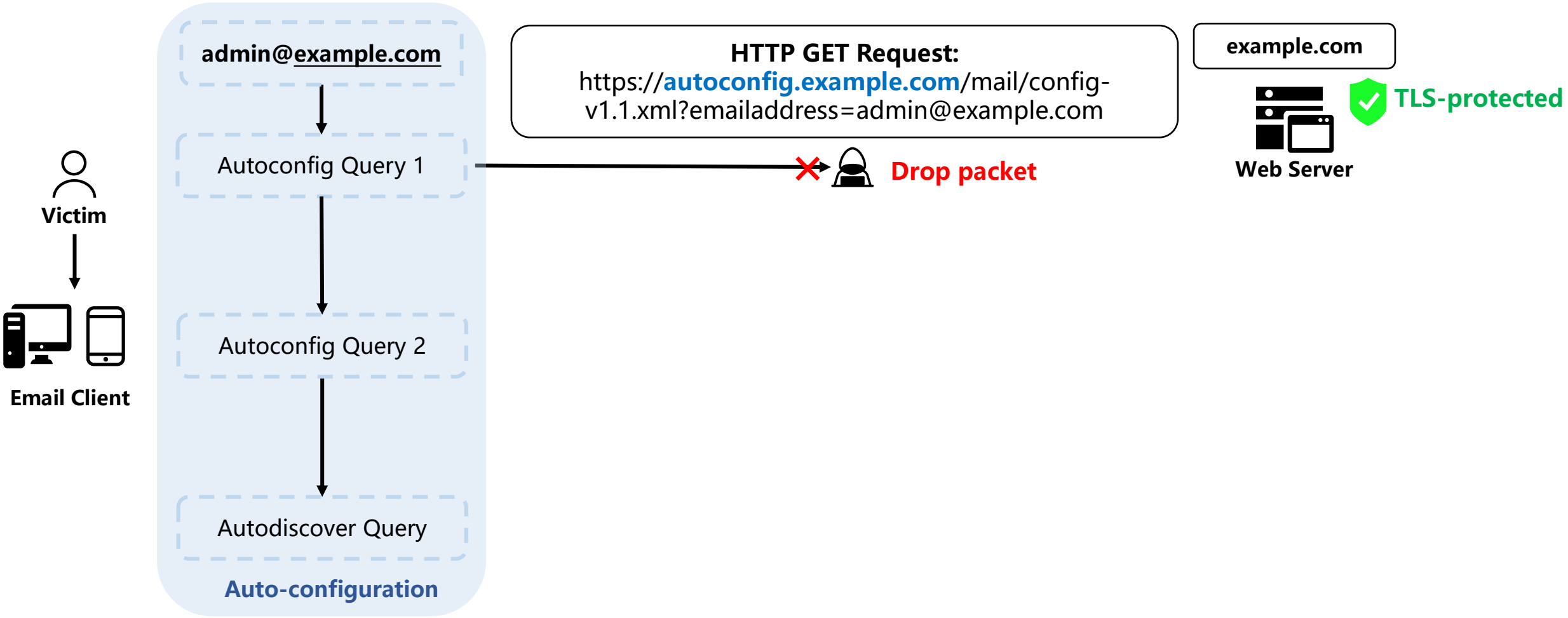
TLS-protected

```
3       <Account>
4       <AccountType>email</AccountType>
5       <Action>settings</Action>
6       <Protocol>
7       <Type>IMAP</Type>
8       <Server>imap.example.com</Server>
9       <Port>143</Port>
10      <SSL>off</SSL>
11      <SPA>on</SPA>
12      </Protocol>
```

autodiscover.xml

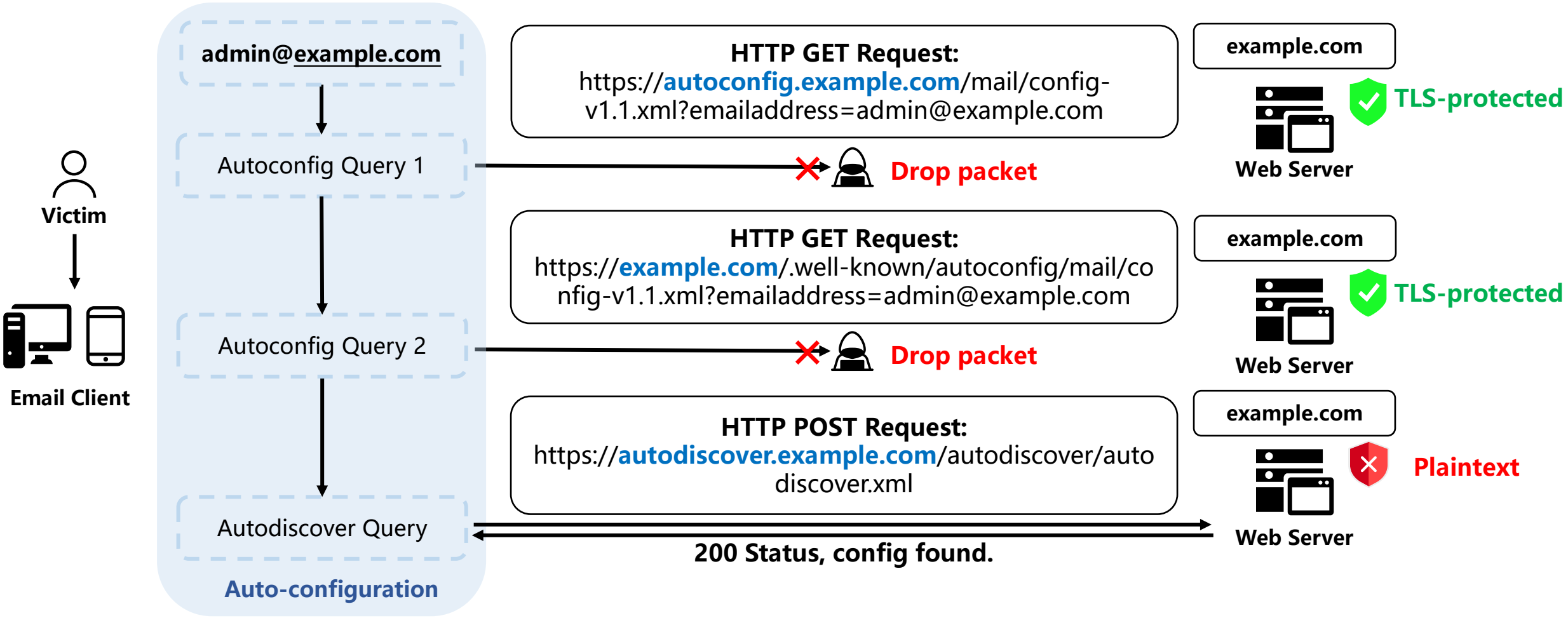
Plaintext

Attack case (A10.1): Inconsistent connection type

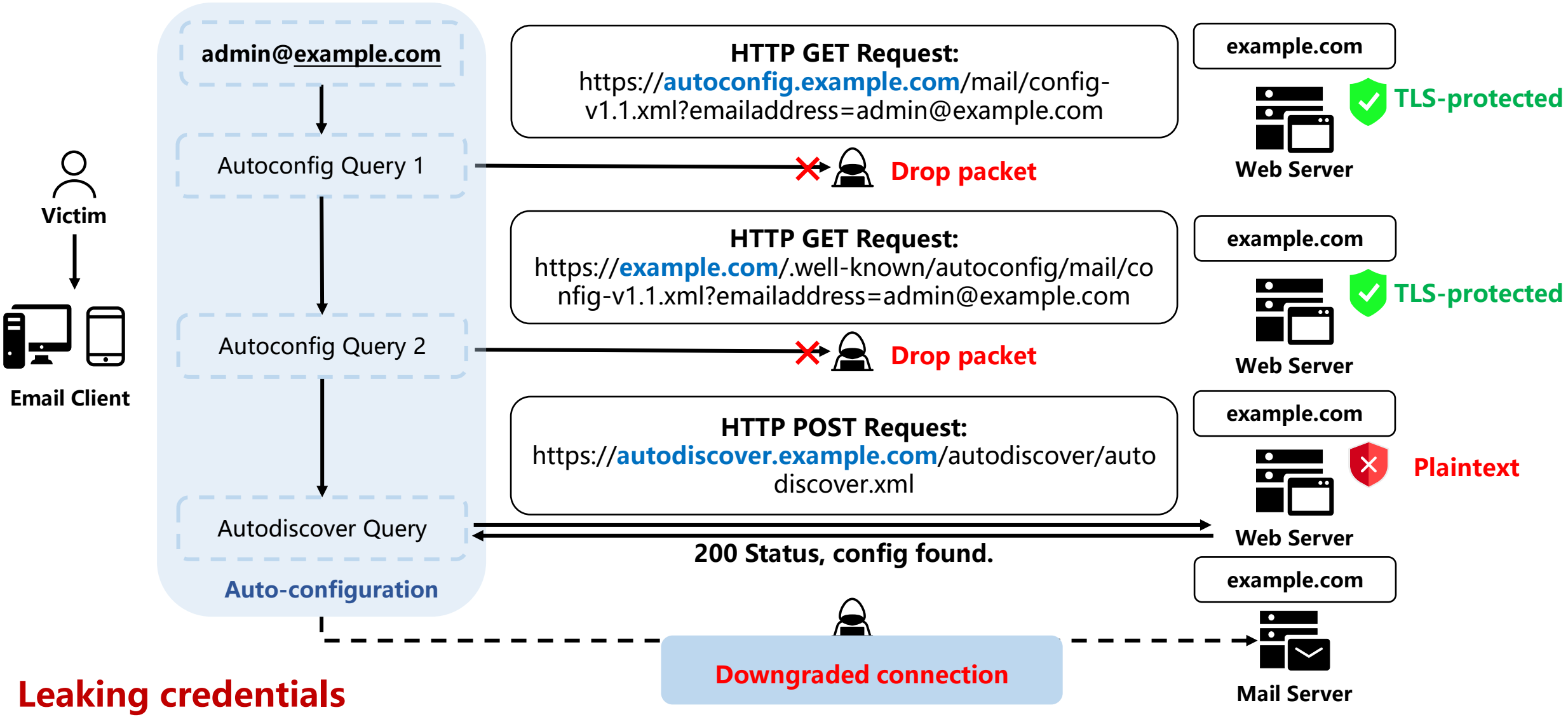


Attack case (A10.1): Inconsistent connection type

Attack case (A10.1): Inconsistent connection type



Attack case (A10.1): Inconsistent connection type



Attack scenarios

- **2/10 Type-I attack scenarios (e.g., plain requests, inadequate eTLD check) and 8/10 Type-II attack scenarios (e.g., plaintext by default).**

Attack goal	Attack scenario	Attacker capability
Type-I: Connecting to attacker-controlled servers	A1: Client requests configuration information in plaintext	Tampering TCP packets
	A2: Client does not enforce eTLD verification	Domain squatting
Type-II: Leaking credentials	A3: Server sets only the plaintext connection type	Sniffing
		Sniffing or hacking STARTTLS[1]
	A4: Client fails to parse and defaults to plaintext	Sniffing
	A5: Client fails to auto-configure and defaults to plaintext	Sniffing
	A6: Client implements Autodiscover inadequately	Sniffing or hacking STARTTLS[2]
	A7: Client prioritizes SRV records incorrectly	
	A8: Client maintains an outdated built-in list.	
	A9: Server prefers insecure connection type	
	A10: Server sets inconsistent connection types	Delaying or dropping packets and, sniffing or hacking STARTTLS[2]

Implementation / Deployment defects

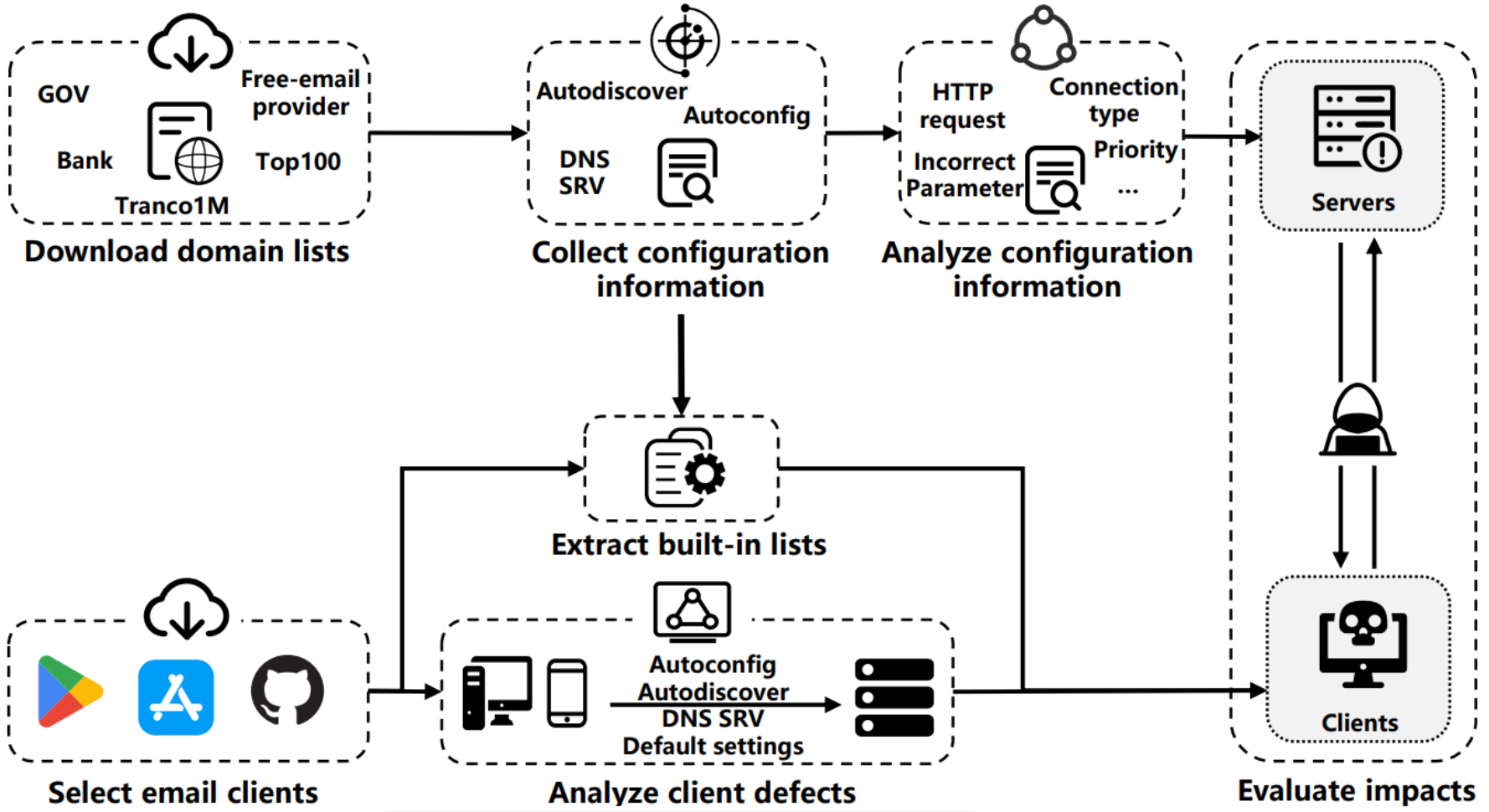
- **17 defects arise from servers or clients.**

Attack case[3]	Applicability[4]	Client defect[5]	Server defect[5]	
			Web	DNS
A1.1	AC	Plain request	Plain response[6]	∅
A1.2	AD	Plain request	Plain response[6]	∅
⁎A1.3	AC/AD	∅	Redirection to HTTP	∅
⁎A2.1	AC	Inadequate eTLD check	∅	∅
A3.1	AC/AD	∅	Plain-only connection	∅
A3.2	SR	∅	∅	Plain or STARTTLS connection
⁎A4.1	AC/AD	Plain fallback on parser error	Incorrect connection type	∅
A5.1	AC/AD/SR/BL	Plain default	∅	∅
⁎A6.1	AD	Ignoring the Encryption element	∅	∅
⁎A7.1	SR	Non-compliant SRV sorting	∅	∅
⁎A8.1	BL	Outdated built-in list	∅	∅
A9.1	AC/AD/SR/BL	∅	∅	Insecure SRV priority
A9.2			Insecure connection priority	∅
⁎A10.1		∅	Inconsistent connection types	

Methodology: Server scanning & client testing

A custom Golang-based crawler and scanned in March 26, 2024

1,053,469 unique domains



Evaluate them separately

29 clients from 5 platforms

A test platform consisted of mail, web, dns servers.

Server results: Deployment status

- The deployment of auto-configuration mechanisms is common.
- However, few domain names have added SRV records.

Unique domains
1,053,469

Support
79,212 (7.52%)

Autodiscover
49,538 (4.70%)

Autoconfig
57,331 (5.44%)

SRV service discovery
11,281 (1.07%)

Server results: Defect evaluation

- **Misconfiguration is prevalent in email auto-configuration deployments.**
  - **49,013/79,212 (61.88%)** misconfigured, including **19** in the Top-1K list.
- **More than half of the servers allow configuration files to be transmitted over a plaintext connection (Type-I).**
  - **43,566 (55.0%)** susceptible to connect to attacker-controlled servers.
- **Having the parameter settings correctly/securely/consistently is non-trivial (Type-II).**
  - **11,824 (14.93%)** face a reduction in connection security, with **2,273** of these could be downgraded to plaintext connections.

Server results: Real-world samples

```
<?xml version="1.0" encoding="utf-8" ?><Autodiscover xm
    <Response xmlns="http://schemas.microsoft.com/excha
        <Account>
            <AccountType>email</AccountType>
            <Action>settings</Action>
            <Protocol>
                <Type>POP3</Type>
                <Server>      home.pl</Server>
                <Port>110</Port>
                <LoginName>info@          /LoginName>
                <DomainRequired>off</DomainRequired>
                <SPA>off</SPA>
                <SSL>off</SSL>
                <DomainRequired>off</DomainRequired>
            </Protocol>
            <Protocol>
                <Type>IMAP</Type>
                <Server>      home.pl</Server>
                <Port>993</Port>
                <DomainRequired>off</DomainRequired>
                <LoginName>info@          /LoginName>
                <SPA>off</SPA>
                <SSL>on</SSL>
                <AuthRequired>off</AuthRequired>
            </Protocol>
```

Insecure connection type preference

```
<Autodiscover xmlns="http://schemas.microsoft.com/exchange
   <Response xmlns="http://schemas.microsoft.com/exchange/
      <Account>
         <Protocol>
            <Type>POP3</Type>
            <Server>      cyber-folks.pl</Server>
            <Port>995</Port>
            <DomainRequired>off</DomainRequired>
            <LoginName></LoginName>
            <SPA>off</SPA>
            <SSL>on</SSL>
            <AuthRequired>on</AuthRequired>
         </Protocol>
         <Protocol>
            <Type>SMTP</Type>
            <Server>      cyber-folks.pl</Server>
            <Port>587</Port>
            <DomainRequired>off</DomainRequired>
            <LoginName></LoginName>
            <SPA>off</SPA>
            <Encryption>STARTTLS</Encryption>
```

"None", "SSL", "TLS", "Auto"

```
         </Protocol>
      </Account>
   </Response>
</Autodiscover>
```

Invalid parameter value

Client results: Mechanisms support & evaluation

- **Most client implementations have flaws that compromise connection security.**
  - **13/29** clients could lead to the victim connecting to an attacker-controlled server (Type-I).
    - **6/13** initiated only plaintext requests (A1.1/A1.2), e.g., **Kmail**.

src/ispdbservice.cpp +7 −11

```
...  ...  @@ -32,11 +32,14 @@ void IspdbService::requestConfig(const KMime::Types::AddrSpec &addrSpec, const S
32   32       QUrl url;
33   33       const QString path = QStringLiteral("/mail/config-v1.1.xml");
34   34       switch (searchServerType) {
     35   +   case IspHttpsAutoConfig:
     36   +       url = QUrl(QStringLiteral("https://autoconfig.") + domain.toLower() + path);
     37   +       break;
35   38       case IspAutoConfig:
36   39           url = QUrl(QStringLiteral("http://autoconfig.") + domain.toLower() + path);
37   40           break;
38   41       case IspWellKnow:
39       -       url = QUrl(QStringLiteral("http://") + domain.toLower() + QStringLiteral("/.well-known/autoconfig") + path);
     42   +       url = QUrl(QStringLiteral("https://") + domain.toLower() + QStringLiteral("/.well-known/autoconfig") + path);
```

Client results: Mechanisms support & evaluation

- **Most client implementations have flaws that compromise connection security.**
  - **13/29** clients could lead to the victim connecting to an attacker-controlled server (Type-I).
    - **Nextcloud Mail used a dot (.) as delimiter** when constructing Autoconfig request (A2.1). **24,149** domains were affected and 54 of which within the Top-10k list.

15 lib/Service/AutoConfig/IspDb.php

```
146 152             }
147 153
148     -       if ($tryMx && ($dns = dns_get_record($domain, DNS_MX))) {
    154 +       if ($tryMx && ($dns = $this->dnsResolver->resolve($domain, DNS_MX))) {
149 155             $domain = $dns[0]['target'];
150 156             if (!($config = $this->query($domain, $email, false))) {
151 157                 [, $domain] = explode('.', $domain, 2);
152     -           $config = $this->query($domain, $email, false);
    158 +           // Only try second-level domains and deeper
    159 +           if (!$this->dnsResolver->isSuffix($domain)) {
    160 +               $config = $this->query($domain, $email, false);
    161 +           }
153 162         }
154 163     }
155 164     return $config;
```

Client results: Mechanisms support & evaluation

- **Most client implementations have flaws that compromise connection security.**
  - **19/29** were susceptible to downgrade to plaintext or STARTTLS (Type-II).
    - **Mailspring** has an outdated built-in list, **last updated 3 years ago**, which prevents it from connecting to the provider (e.g., alice.it) using the latest configuration.
    - Besides, at least **71** domains in ISPDB had outdated configurations.

- **UI notifications as a last line of defense are almost absent.**
  - All clients, except Thunderbird and K-9 Mail, had at least one UI-related defect.
    - **21** clients **did not prompt users to confirm the results.**
    - Leaving victims unaware of their connection to an attacker-controlled server.

Client results: Mechanisms support & evaluation

Table VII: Evaluation results of 29 email clients.

Client[1]	Auto-configuration Support[2]					Default Port [P/S][3]		Defect	UI Notification[4]			
	Autoconfig	Autodiscover	DNS SRV	Built-in list	Guess	Incoming	Outgoing		UC	W_P	W_{AD}	W_{SR}
Windows												
Postbox (7.0.60)	●$_P$	○	○	●$_I$	●	143 S	587 S	A1.1, A8.1	✓	✗		
Delta Chat (1.42.1) ∗	●	●	○	●$_I$	●	993 S	465 S	A6.1, A8.1	✗			
Outlook (16.0.10406.20006)	○	●	○	●	●	143 P	25 P	A4.1, A5.1	✗		✓	
Mailbird (3.0.6.0)	○	◐	○	●	●	–	–	A1.2	✗			
eM Client (9.2.2157)	○	●	○	●	●	143 S	587 S		✗		✗	
The bat! (11.0.3.1)	○	○	○	●	●	143 P	25 P	A5.1	✓	✗		
Linux												
Claws Mail (4.2.0git36) ∗	○	○	●	○	○	143 P	25 P	A5.1	✓	✗		✗
Thunderbird (115.6.0) ∗	●	●	○	●$_I$	●	143 P	587 P	A1.1, A4.1, A5.1, A6.1, A8.1	✓	✓	✓	
Kmail (5.24.4) ∗	●$_P$	○	○	●$_I$	○	993 S	25 S	A1.1, A4.1, A8.1	✓	✗		
Evolution (3.50.3) ∗	●	○	●	●$_I$	○	993 S	465 S	A1.1, A4.1, A8.1	✓	✗		✗
Nextcloud Mail (3.5.3) ∗	●	○	○	●$_I$	○	993 S	587 S	A1.1, A2.1, A4.1, A8.1	✗			
Geary (44.1) ∗	●	○	○	●$_I$	●	993 S	465 S	A4.1, A8.1	✗			
Android												
FairEmail (1.2149a) ∗	●	●	●	●$_I$	●	993 S	465 S	A1.1, A6.1, A7.1, A8.1	✗			✗
Nine (4.9.5e)	●$_P$	●	●	●$_I$	●	993 S	465 S	A1.1, A1.2, A6.1, A8.1	✗		✗	✗
MailTime (4.1.5.1218)	●$_P$	◐	○	●$_I$	●	993 S	465 S	A1.1, A8.1	✗		✗	
K-9 Mail (6.714) ∗	●	○	○	●$_I$	○	993 S	465 S	A1.1, A8.1	✓	N/A		
Spark Mail (3.7.2)	●	○	○	●$_I$	●	993 S	587 S	A8.1	✗			
ProfiMail Go (4.32.00)	●	○	○	●$_I$	●	143 P	25 P	A4.1, A5.1, A8.1	✗			
Maildroid (5.22)	○	○	○	●$_I$	●	143 P	25 P	A5.1, A8.1	✗			
iOS												
myMail (14.71.0)	○	○	○	●	●	993 S	465 S		✗			
iOS Mail (17.1)	○	○	○	●	○	993 S	587 S		✗			
Edison Mail (1.53.14)	○	◐	○	●	●	993 S	587 S		✗		✗	
Gmail (6.0.240225)	○	○	○	●	○	993 S	465 S		✗			
Mailbus (3.3.11)	○	◐	○	●	●	993 S	465 S	A1.2	✗			
AltaMail (8.2.5)	○	●$_P$	○	●	●	143 S	25 S	A1.2, A6.1	✗			
MacOS												
Apple Mail (13.5.2)	○	○	○	●	○	993 S	465 S		✗			
Airmail (5.7)	○	○	○	●	●	993 S	465 S		✗			
Mailspring (1.13.3) ∗	○	○	○	●	○	993 S	465 S	A8.1	✓	✗		
Spike (3.8.0)	○	◐$_P$	●	●	○	993 S	587 S	A1.2	✗	✗	✗	✗

[1] ∗ Open-source client.
[2] ○ - Not support. ● - Supported. ◐ - Support Autodiscover for Exchange only. P indicates the client only initiates plaintext requests and I indicates the client queries the ISPDB.
[3] P - Default to plaintext. S - Encrypted connection through implicit TLS or STARTTLS. Mailbird does not provide a default connection type and port, requiring the user to enter manually.
[4] UC - User confirmation. W_P - Plaintext warning. W_{AD} - Autodiscover redirect warning. W_{SR} - SRV FQDN warning. ✓ means a UI notification, and ✗ means no UI notification. K-9 Mail requires the user to enter configuration parameters manually when the configuration information retrieved contains a plaintext connection type.

13 Autoconfig
12 Autodiscover
5 SRV service discovery
28 Built-in lists
19 heuristic guessing

22/29 flawed

27/29 have UI-related defects

Mitigation

- **Enforcing secure connections, especially in clients.**
  - Client should take the lead to enforce TLS.
  - Administrators should migrate from plaintext or STARTTLS to implicit TLS.
- **Checking and updating configuration regularly.**
  - Administrators should regularly check and update configurations.
  - Developers should ensure that the built-in lists used contain the latest configurations.
  - Tool: **https://github.com/emailconfigtest/mailconfig**
- **Implementing professional clients.**
  - Both the effectiveness and security should be considered.

Summary

- The first systematic analysis of the security threats in email auto-configuration mechanisms.
- Extensive measurements to evaluate the real-world impact of these threats.
- Widespread flaws in server deployments and client implementations.

中国科学技术大学
University of Science and Technology of China

清華大學
Tsinghua University

北京航空航天大學
BEIHANG UNIVERSITY

Automatic Insecurity: Exploring Email Auto-configuration in the Wild

Thank you for listening, any questions?

Contact: Shushang Wen, sswen@mail.ustc.edu.cn

University of Science and Technology of China

NDSS SYMPOSIUM/2025

Internet Society

Backup: Other mechanisms, heuristic guessing

- **Heuristically determine mail server configuration.**
  - For example, Thunderbird prefixes the domain name with a relevant protocol (e.g., "imap", "pop3" or "smtp").

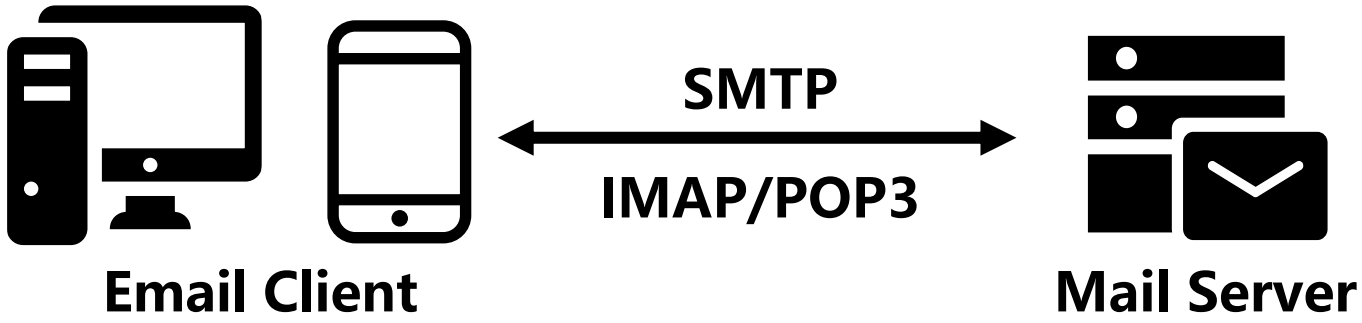
```
IncomingHostDetector.prototype = {
  __proto__: HostDetector.prototype,
  _hostnamesToTry(protocol, domain) {
    var hostnamesToTry = [];
    if (protocol != POP) {
      hostnamesToTry.push("imap." + domain);
    }
    if (protocol != IMAP) {
      hostnamesToTry.push("pop3." + domain);
      hostnamesToTry.push("pop." + domain);
    }
    hostnamesToTry.push("mail." + domain);
    hostnamesToTry.push(domain);
    return hostnamesToTry;
  },
  _portsToTry: getIncomingTryOrder,
};
```

```
OutgoingHostDetector.prototype = {
  __proto__: HostDetector.prototype,
  _hostnamesToTry(protocol, domain) {
    var hostnamesToTry = [];
    hostnamesToTry.push("smtp." + domain);
    hostnamesToTry.push("mail." + domain);
    hostnamesToTry.push(domain);
    return hostnamesToTry;
  },
  _portsToTry: getOutgoingTryOrder,
};
```

Backup: Other mechanisms, default setting

- **Default setting when email auto-configuration fails.**
  - If none of the above mechanisms return settings, most clients preset a default value for the connection type.

*Determines if the client-to-mail server connection is **plaintext** or **TLS-protected** (via STARTTLS or implicit TLS).*



Backup: Definition of the configuration file

Element	Value	Meaning
hostname	custom	The name of the mail server
port	custom	Customized by the server, typically well-known ports, e.g., 993, 995, 465 etc.
socketType	plain, starttls, ssl	Plaintext, or establish an encrypted connection by STARTTLS or SSL.
authentication	password-cleartext, password-encrypted, NTLM, GSS-API, client-IP-address, TLS-client-cert, OAuth2, none	Authentication methods

Element	Value	Meaning
Server	custom	The name of the mail server
Port	custom	Customized by the server, typically well-known ports, e.g., 993, 995, 465 etc.
SSL	on, off	Whether to establish an encrypted connection, default is 'on'
SPA	on, off	Whether secure password authentication is required, default is 'on'
Encryption	none, ssl, tls, auto	If present, overrides the SSL element. 'none' represents no encryption is uesed. 'ssl' and 'tls' stand for Secure Sockets Layer (SSL) or Transport Layer Security (TLS) is used, respectively, where SSL is superseded by TLS. 'auto' represents using the most secure encryption that both client and server support.
DomainRequired	on, off	Whether the domain is required for authentication

[1] "Mail Autoconfig," Internet-Draft draft-bucksch-autoconfig-00, https://datatracker.ietf.org/doc/draft-bucksch-autoconfig/00/
[2] "[MS-OXDSCLI]: Autodiscover HTTP Service Protocol," https://msopenspecs.azureedge.net/files/MS-OXDSCLI/%5bMS-OXDSCLI%5d-210817.pdf