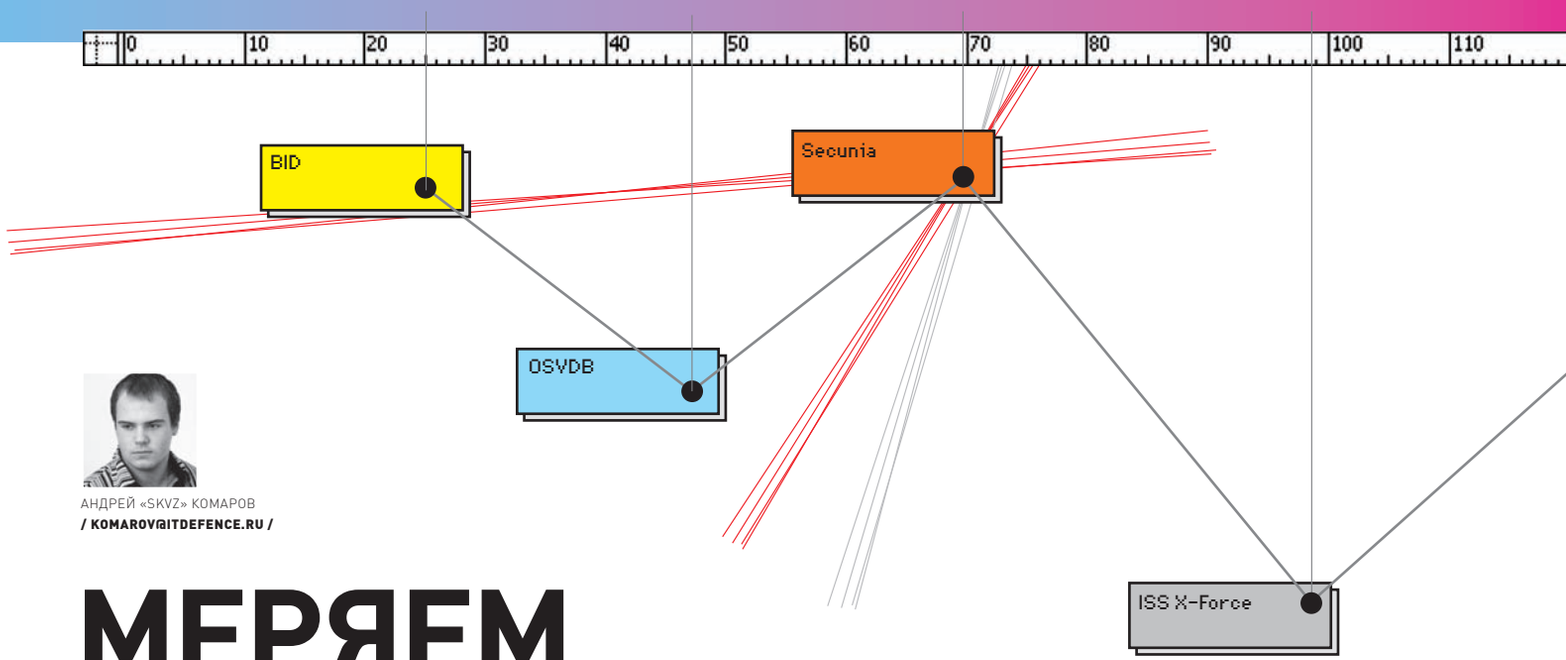





АНДРЕЙ «SKVZ» КОМАРОВ
/ KOMAROV@ITDEFENCE.RU /

# МЕРЯЕМ УЯЗВИМОСТИ

## КЛАССИФИКАТОРЫ И МЕТРИКИ КОМПЬЮТЕРНЫХ БРЕШЕЙ

Ежедневно сотнями хакеров обнаруживаются тысячи уязвимостей, — после чего взламывается куча сайтов, и детали багов выкладываются в багтрак на всеобщее обозрение. Наверняка, ты читал подобные обзоры и замечал, что каждый баг определенным образом классифицируется. Что собой представляет измерение уязвимости, по каким критериям производится и на кой черт это вообще нужно знать? Ответы ты найдешь в этой статье.



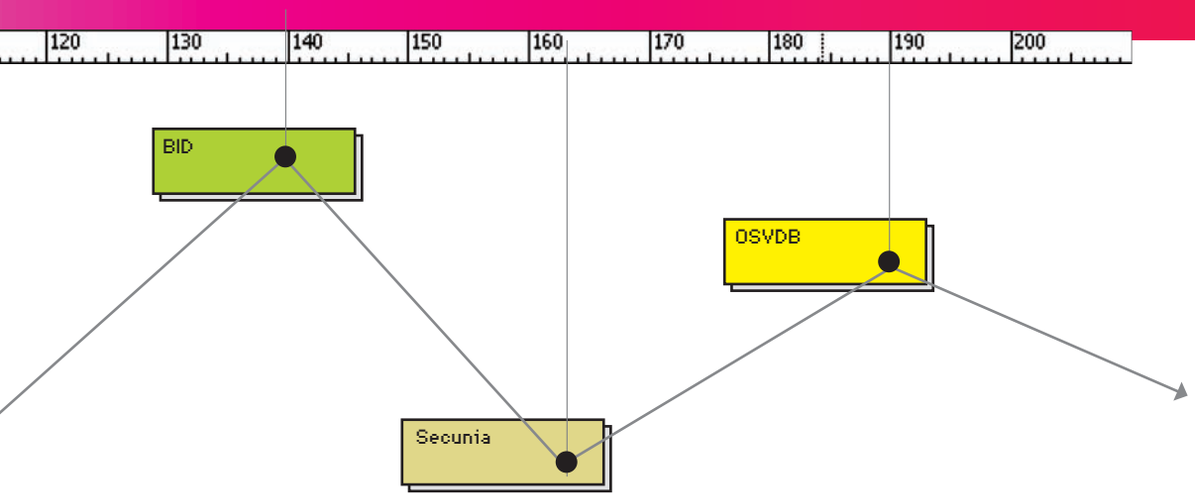
**«Общепринятых систем по классификации брешей в нашей стране не существует»** — эту фразу я поставлю во главу угла. Продвинутым государством в этом плане стали США. Там ведут несколько классификаций и активно используют их как в образовательном процессе, так и в технологиях. Одной из самых известных систем классификации является CVE, которая курируется компанией NCSD (National Cyber Security Division) при одном из Министерств США. Рассмотрим эту систему подробнее.

### CVE (COMMON VULNERABILITIES AND EXPOSURES)

По сути, CVE — это «словарь» известных уязвимостей, имеющий строгую характеристику по описательным критериям, что отличает его, скажем, от Bugtrack-ленты. Полностью CVE можно отыскать в Национальной Базе Уязвимостей США (NVD — nvd.nist.gov) или на официальном сайте (cve.mitre.org/data/downloads). Причем, распространяется база в нескольких форматах: xml, html, csf, xsd schema. Из-за такой доступности, открытости и удобства к базе CVE часто обращаются сами разработчики различного ПО (в первую очередь, нацеленного на рынок информационной безопасности). Общий вид записи CVE выглядит примерно так:

```
CVE ID, Reference и Description.
```

ID записывается с указанием кода и порядкового номера, например «CVE-1999-03». В поле Reference записываются различного рода ссылки на патчи, рекомендательного рода документы или комментарии разработчика. Description отвечает за описание самой уязвимости. Короче, CVE — система широкого профиля и никоим образом не сосредотачивается только на клиентских уязвимостях или, скажем, исключительно на WEB-протоколе. Изначально она задумывалась как единый стандарт идентификации уязвимостей, который должен охватывать несколько звеньев информационной системы: систему поиска и обнаружения брешей (например, сканер безопасности), антивирусное ПО, а также исследуемое ПО. Как появилась идея ее создания? Многие компании занимаются поиском брешей в различных продуктах на основе политики (не)разглашения информации и взаимодействия с производителями. Как-то, при исследовании одного из продуктов десятью различными компаниями, одной и той же уязвимости были присвоены абсолютно разные названия. После выявления сей вопиющей несправедливости было принято соглашение о едином стандарте. Тогда же компания MITRE Corporation (mitre.org) предложила решение, независимое от различных производителей средств поиска уязвимостей, и взяла на себя ответственность за его воплощение. Нельзя сказать, что после этого все баги стали упорядоченными. Разработчики продолжают активно развивать самостоятельные начинания. Часть из них имеют платную подписку, и антивирусные компании частенько обращаются




к ним и добавляют соответствующие сигнатуры в свои продукты. Стоимость такой годовой подписки составляет около $5000 и выше.

### BID

Эта классификация присутствует исключительно на портале Securityfocus (используется в ленте securityfocus.com/vulnerabilities). Одна из отличительных особенностей BID — совместимость с CVE. Условно говоря, найденная уязвимость в BID имеет ссылку на номер CVE и, соответственно, равнозначна по информации. У системы есть ряд описательных свойств — например, класс, возможность локального или удаленного исполнения и т.п. В будущем ты убедишься, что этих параметров недостаточно для полной характеристики, но, тем не менее, BID дает разработчику вполне наглядную информацию о выявленной бреши.

### OSVDB

Название расшифровывается примерно как: «Открытая база данных уязвимостей». Все просто и со вкусом. Классификация создана тремя некоммерческими организациями. Двести волонтеров со всего мира активно участвуют в ее наполнении. Среди прочего присутствуют: локация эксплуатации (сетевой доступ/локальный доступ) и импакт (ущерб от уязвимости, воздействие на какую-либо часть целевой информационной системы).

### SECUNIA

Эта датская компания, лента уязвимостей которой доступна по адресу secunia.com, уже заработала себе достаточно славы. Не сказать, чтобы их портал внес какую-то особую, добавочную классификацию, но именно он предлагает услуги платной подписки на базу уязвимостей.

### ISS X-FORCE

ISS затрагивает все перечисленные выше критерии, но вдобавок описывает бизнес-импакт, а именно — материальный ущерб, который может повлечь за собой угроза эксплуатации. Например, баг «Microsoft Excel Remote Code Execution», нацеленный на компьютер сотрудника банка или предприятия, способен привести к краже важных документов, ущерб от разглашения которых может исчисляться миллионами. Оценить урон от различных видов атак можно, ознакомившись с одним из ведущих блогов в русскоязычном сегменте о security-бричах и утечках — Perimetrix (securitylab.ru/blog/company/Perimetrix_blog). Также в системе присутствует качественно новая черта — переход к метрикам безопасности для описания свойств уязвимости. Для этого используется общая система подсчета рисков уязвимостей CVSS версии 2. Она представляет собой шкалы, на основе которых выставляются баллы. Система метрик была придумана для разделения приоритетов над исправлением уязвимостей. Каждая шкала относится к определенному смысловому разделу, который называется метрикой. В CVSS v.2 их три: базовая метрика, временная метрика и контекстная метрика. Хакеров заинтересует только первая.

### БАЗОВАЯ МЕТРИКА

Нередко на солидных порталах по безопасности можно увидеть фразу — «CVSS Base Score = 9.2». Как это понимать? Параметр вычисляется по специальной формуле:

```
BaseScore = round_to_1_decimal(((
0.6*Impact)+(0.4*Exploitability)-
1.5)*f(Impact))
```

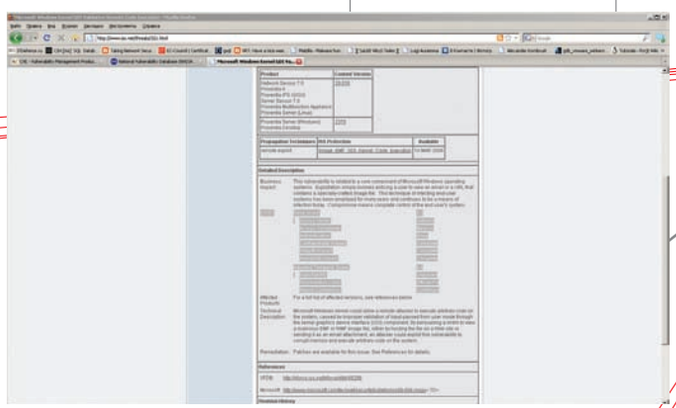
— плюс еще несколько. Все эти формулы можно найти по адресу first.org/cvss. Чтобы все стало понятнее, рассмотрим пример. Задан вектор уязвимости базовой метрики вида: «AV:N/AC:L/Au:N/C:N/I:N/A:C». Все красуется на странице описания, но ты абсолютно не можешь расшифровать эти иероглифы! Я тебе помогу. Итак, расшифровываем по порядку. **Access Vector: Network** — возможность доступа к объекту исключительно через сеть. Для эксплуатации уязвимости злоумышленник должен обладать доступом к уязвимому ПО, причем этот доступ ограничен только величиной сетевого стека. Локального доступа или доступа из

### Недокументированные уязвимости

В настоящее время эксперты мозгуют над включением вектора «недокументированные уязвимости» в одну из метрик. Этот параметр имеет высокое значение. В недалеком будущем мы сможем лицезреть строку undercover vulnerabilities — «вероятно, возможные к исполнению». На сайте, посвященном развитию метрик информационной безопасности (securitymetrics.org/content/Wiki.jsp), вывешено публичное обращение по поводу того, как и каким образом исчислять этот параметр. Все желающие могут отправить туда свои предложения.

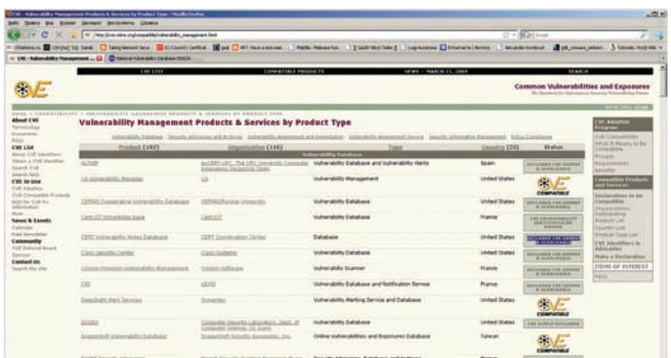
### Политика разглашения информации об уязвимости

Это соглашение имеет ряд нюансов. Например, хакер, обнаружив уязвимость, ищет контакты, чтобы направить соответствующий запрос производителю. Если по истечении пяти дней производитель отмалчивается, вводит в заблуждение своих пользователей какими-то способами или некорректно вступает в диалог, то ему отправляется повторное письмо. Выжидаются еще пять рабочих дней, после чего баг-хантер вправе помещать описание о баге на собственном ресурсе или в публичные багтраки. При этом в письме требуется оговорить и согласовать дату публикации, чтобы производитель успел выпустить обновление или советы по защите от эксплуатации. Важно отметить, что если стороннее третье лицо опубликовало данные об эксплуатации найденной тобой уязвимости, ты можешь смело постить ее подробности без согласования с кем-либо. Вот такая арифметика.

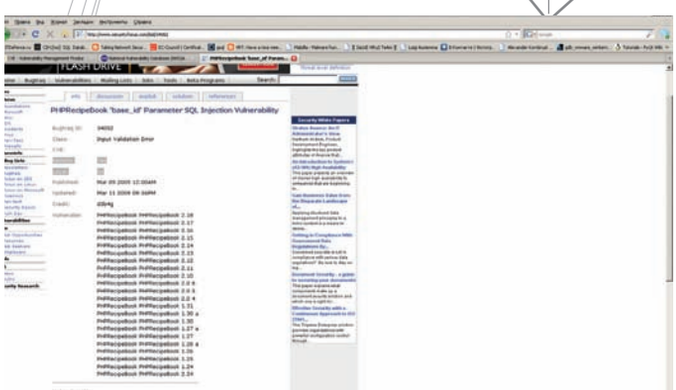


Подсчитанное значение Base Score для уязвимости в Microsoft Windows Kernel GDI

Все самое интересное и вкусное распространяется только платно



Существует очень много видов систем и баз для учета уязвимостей. Многие из них совместимы с CVE



BID указывает лишь несколько характеристик о свойствах уязвимости

**info**

Истинные корни создания единой классификации багов и их контроля — это Unix Known Problem List, Internal Sun Microsystems Bug List, каталоги служб реагирования на компьютерные инциденты CERT ранних версий.

соседней сети не требуется. Такие уязвимости часто называют эксплуатируемыми удаленно. Примером такой сетевой атаки служит переполнение буфера RPC.

**Access Complexity: Low** — сложность доступа к ресурсу: низкая. Для эксплуатации специальных условий и особых обстоятельств не требуется — все стандартно, шаблонно, общедоступно.

**Authentication: None** — для эксплуатации не нужна авторизация. Например, если бы это был сервис, который требует предварительной авторизации по какой-нибудь мудреной схеме (смарт-карты, ключи, токены), то значение этого вектора было бы другим.

**Confidentiality Impact: None** — влияние на разглашение критичной информации. Integrity Impact: None — нарушение целостности. Понятие «целостность» связано с достоверностью и точностью информации. Если бы у злоумышленника была возможность модификации файлов, изменения области исполнения файлов, то мы бы поставили здесь C (полное) или P (частичное, от «partial»).

**Availability Impact: Complete** — атаки, потребляющие пропускную способность сети, циклы процессора или дисковое пространство, которые влияют на доступность системы. Если эксплуатация уязвимости вызывает отказ в обслуживании, то Availability Impact имеет значение «Complete».

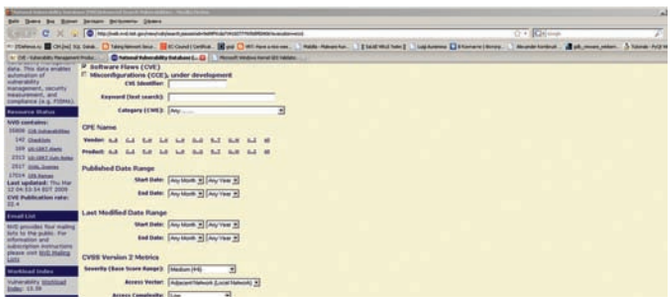
## ВРЕМЕННАЯ МЕТРИКА

Более глубоким анализом занимаются временные и контекстные метрики. Дело в том, что описанные векторы базовой метрики со временем не меняются. Они постоянны и могут характеризовать уязвимость по назначению и опасности. А какие критерии могут изменяться с течением времени? Представь, что ты нашел критическую уязвимость и уведомил разработчика. Временной интервал исправления уязвимости в таком случае имеет значение, да и к тому же, сам изменяется во времени (это может быть день, час, либо производитель вообще никак не отреагирует). Или ситуация, когда твой друг написал боевой эксплойт на недавнюю уязвимость «нулевого дня». Как долго этот код будет актуален? Он ускорит риск эксплуатации, следовательно, должен учитываться при ее описании. Доступна ли будет его технология к завтрашнему дню? На все эти вопросы отвечают временные метрики. Рассмотрим некоторые их векторы.

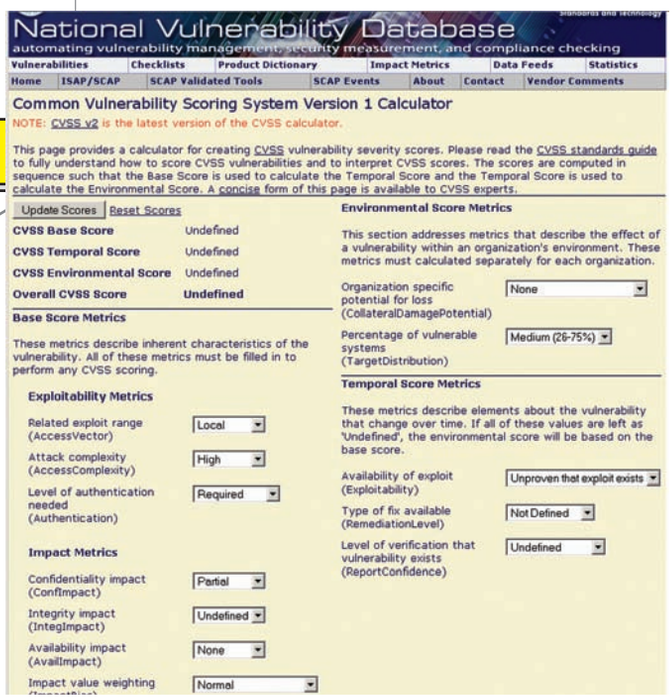
**Exploitability (E)** — возможность эксплуатации. Пожалуй, один из важнейших критериев. Речь идет конкретно о доступности средства (кода, эксплойта, технологии), которое успешно работает. Важно учитывать и то, что доступный эксплойт можно использовать далеко не всегда. Используемые описательные флаги: U (недоступен или непроверен), Proof-of-Concept (POC — опубликована наглядная демонстрация уязвимости), F (функциональный, и рабочий эксплойт у тебя в руках), H («high risk» всей темы, чаще всего характерен для червей или для уязвимостей с широко популярным описанием), ND (без разницы, вектор метрики не влияет ни на что существенное, поэтому учитывать его не надо). **Remediation Level (RL)** — уровень исправления. Голос уязвимости услышал весь свет, вот только как поступят разработчики? Порой они просто молчат, потому что их уже не осталось в живых (простите, за цинизм и черный юмор), а иногда абсолютно сторонние организации и неофициальные источники начинают заботиться о безопасности на первый взгляд чужих продуктов и оперативно писать заплатки.

**Report Confidence (RC)** — степень достоверности отчета. Сколько слухов и разговоров крутится вокруг! Банальный

Вывод характеристик с сайта osvdb.org. Не всегда авторам известно о наличии боевого кода в природе, что затемняет подробности уязвимости



На официальном сайте Национальной базы данных уязвимостей есть возможность поиска по критериям CVSS




Если ты заметил, мы обращались к новой редакции CVSS (2). Существует еще и первая, облегченная версия, не учитывающая многих параметров. Тем не менее, она пригодна к использованию и намного проще. Посчитать Base Score для нее можно с помощью сервиса на офсайте NVD

пример: человек написал информацию якобы о рабочей критической уязвимости. А на деле оказалось, что это программный дефект и ничего существенного собой не представляет. Подтверждена ли уязвимость экспертами или же это просто проделки хакерских слухов? Ответ на этот вопрос даст вектор Report Confidence. Параметры всех указанных векторов градируются вариантами «да/нет/возможно».

## КОНТЕКСТНАЯ МЕТРИКА

Эти группы векторов отражают влияние на среду пользователя и изучают поведение после эксплуатации уязвимости. Как правило, метрика используется в качестве дополнения к базовой. **Collateral Damage Potential (CDP)** — вероятность нанесения косвенного ущерба. Описывает экономические или технические потери. Скажем, нам встретится уязвимость, приводящая к DoS-атаке. После ее эксплуатации часть сетевого оборудования перегревается, не справляясь с работой, и выходит из строя. Но при этом ущерб оказывается незначительным из-за низкой стоимости устройства и его расположения (вне защищаемых и важных объектов). **Target Distribution (TD)** — плотность целей. Влияет ли уязвимость только на одну цель, либо с ее помощью можно поработить огромное число машин? Если это стендовое показательное выступление, лабораторный практикум или эксплуатация на машине, изолированной от других, то значение этого вектора равно нулю.

## ИСПОЛЬЗОВАНИЕ КЛАССИФИКАТОРОВ В СКАНЕРАХ

Современные автоматизированные аудиторы принято затачивать под какую-либо конкретную базу знаний. Во-первых, это престижно, во-вторых — полезно. К примеру, при подготовке к аттестации по одному из современных стандартов (NERC-CIP, PCI, FISMA, GLBA или HIPAA) администратору предоставляется возможность получить шаблонный отчет, соответствующий документу, издаваемому аудитором. Я встречал такое в современных сканерах беспроводной безопасности, типа AirMagnet, а также дорогих коммерческих сканерах вроде ISS Security Scanner. Порой сканеры безопасности прибегают к использованию собственного разделения брешей по ID. Подобная практика применяется в Nessus, который таки сменил лицензию на полукоммерческую.

## ОТДЕЛЬНЫЕ КЛАССИФИКАЦИИ

Подчас в Сети можно заметить абсолютно самопальные классификации, вроде Common Criteria Web Application Security Scoring (CCWAPSS) 1.1. Естественно, большого веса такая система не имеет, потому что составлять ся она должна реальными экспертами, которые понимают суть проблемы.

### Список „междусобной" совместимости систем классификаций

CVE: ISS, BID, Secunia, SecurityTracker, OSVDB
BID: CVE, Bugtraq, ISS, Secunia, SecurityTracker, OSVDB
ISS: CVE, BID, Secunia, SecurityTracker, OSVDB
Secunia: CVE, OSVDB
SecurityTracker: CVE, OSVDB, Nessus
Nessus: CVE, BID, OSVDB
OSVDB: CVE, BID, Secunia, SecurityTracker, ISS, Nessus, Snort

## ТАК ЛИ ОНО ВСЕ ВАЖНО?

Безусловно, к делу следует подходить без фанатизма. В первую очередь, подобные системы классификации нацелены на экспертное звено либо специалистов, которые заботятся о своевременном устранении брешей. Но, на мой взгляд, каждый уважающий себя хакер должен знать и понимать общепринятые классификации уязвимостей, разбираться в метриках и их векторах, чтобы четко и ясно представлять формулу оценки всех недавно взломанных им ресурсов.